

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **ZAVRŠNI RAD**

**Mario Svačina**

**Zagreb, 2014.**

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **ZAVRŠNI RAD**

Mentor:

Prof. dr. sc. Dubravko Majetić

Student:

Mario Svačina

Zagreb, 2014.

Izjavljujem da sam završni rad izradio samostalno, služeći se literaturom i znanjem stečenim tijekom studija. Zahvaljujem se mentoru prof. dr. sc. Dubravku Majetiću na pomoći i sugestijama prilikom same izrade završnog rada. Također, zahvaljujem se svojoj djevojci, obitelji i prijateljima na podršci koju su mi pružili tijekom ovog dijela studija.

# SADRŽAJ

Popis slika.....	III
Popis tablica.....	IV
Summary.....	VI
Ključne riječi.....	VII
Key words .....	VIII
Popis važnijih oznaka.....	IX
Popis mjernih jedinica .....	IX
1. Uvod .....	1
1.1. Umjetne neuronske mreže .....	1
1.1.1. Biološki neuron.....	2
1.1.2. Sličnosti i razlike mozga i računala .....	3
1.1.3. Umjetni neuron .....	3
1.1.4. Podjela umjetnih neuronskih mreža .....	5
1.1.5. Učenje umjetnih neuronskih mreža .....	5
1.2. Matlab.....	6
1.2.1. Programiranje u Matlab-u .....	6
1.2.2. Grafičko korisničko sučelje (eng. Graphic User Interface GUI).....	7
1.2.3. GUIDE (Graphical User Interface development enviroment – Okolina za izradu grafičkog korisničkog sučelja).....	7
2. Statička unaprijedna neuronska mreža s povratnim prostiranjem pogreške (eng. EBP- Error-Back Propagation) .....	8
2.1. Perceptron .....	8
2.1.1. Učenje Perceptrona .....	8
2.2. Višeslojne statičke neuronske mreže .....	9
2.2.1. Model statičkog neurona .....	9
2.2.2. Model statičke neuronske mreže .....	11
2.2.3. Učenje povratnim rasprostiranjem pogreške.....	12
2.2.4. Unaprijedna faza učenja.....	13

2.2.5. Povratna faza učenja .....	14
2.2.5.1. Povratna faza po uzorku.....	14
2.2.5.2. Povratna faza po skupu .....	15
2.2.6. Ubrzanje iterativnog učenja .....	15
2.2.6.1. Zamah prvog reda .....	16
2.2.6.2. Zamah drugog reda .....	16
2.2.7. Promjena težina izlaznog sloja .....	16
2.2.8. Promjena težina sakrivenog sloja .....	18
2.2.9. Ocjena točnost algoritma učenja .....	20
3. Statička unaprijedna neuronska mreža za raspoznavanje osoba.....	22
3.1. Topologija mreže .....	22
3.2. Karakteristike osoba.....	22
3.3. Unošenje parametara učenja i učenje mreže .....	25
3.4. Testiranje mreže sa 6 osoba .....	26
3.5. Prikaz rezultata različitih načina učenja mreže.....	27
4. Zaključak .....	30
5. Literatura .....	31

## **Popis slika**

Slika 1. 1: Skica biološkog neurona.....	2
Slika 1. 2: Umjetni neuron.....	4
Slika 2. 1: Model statičkog neurona .....	10
Slika 2. 2: Prikaz nekih aktivacijskih funkcija .....	11
Slika 2. 3: Model statičke unaprijedne neuronske mreže .....	12
Slika 3. 1: Prikaz grafičkog sučelja za unos karakteristika osoba.....	25
Slika 3. 2: Prikaz grafičkog sučelja za unos parametara učenja .....	26
Slika 3. 3: Prikaz grafičkog sučelja za testiranje mreže .....	27

## **Popis tablica**

Tablica 1: Prikaz rada mreže sa koeficijentom učenja $\eta=0,01$ .....	28
Tablica 2: Prikaz rada mreže sa koeficijentom učenja $\eta=0,04$ .....	29

## Sažetak

Umjetne neuronske mreže su dio šireg područja zvanog umjetna inteligencija. Cilj umjetne inteligencije, pa tako i umjetnih neuronskih mreža je napraviti sustave koji se ne oslanjaju isključivo na determinističke matematičke postupke te kojima nije potrebno da svi ulazi u sustav budu potpuno točni. Pomoću umjetnih neuronskih mreža to je u potpunosti moguće. U ovom radu je prikazan rad umjetne neuronske mreže sa povratnim prostiranjem pogreške u svrhu prepoznavanja osoba. Također, pokazano je kako neuronska mreža uči te na koji način se mreža kasnije testira. Rad i učenje neuronske mreže te unošenje osoba i njihovih karakteristika se izvodi preko Matlab-ovog grafičkog sučelja. Grafikom je omogućeno korištenje umjetne neuronske mreže za prepoznavanje osoba. Grafika je napravljena na način da je u potpunosti prilagođena korisniku te je omogućeno korištenje neuronske mreže i osobama koje ne znaju samu suštinu rada umjetnih neuronskih mreža.



## **Summary**

Artificial neural networks are part of a larger area known as artificial intelligence. The aim of artificial intelligence, including artificial neural networks, is to create systems which do not rely solely on deterministic mathematical procedures and systems which do not require all inputs to the system to be completely accurate. Artificial neural network makes it entirely possible. This paper deals with the functioning of artificial neural networks with error-back propagation whose goal is to identify persons. Moreover, it shows network's methods of learning and testing of the results. Operations and learning of artificial neural network, as well as the input of persons' characteristics, are performed through MATLAB's graphical user interface. The recognition of persons was enabled by using graphics. Graphics made the interface completely adjusted to the user. Most importantly, it made the using of artificial neural networks available to the users which are not familiar with their functioning.

## **Ključne riječi**

umjetni neuron

umjetne neuronske mreže

težinski faktori

povratno rasprostiranje pogreške

klasifikacija

zamah prvog reda

zamah drugog reda

## **Key words**

artificial neuron

artificial neural networks

weight factors

error-back propagation

classification

first order momentum

second order momentum

## **Popis važnijih oznaka**

**Z**- matrica ulaznih neurona

**D**- matrica izlaznih neurona

**V**- matrica težina sakrivenih nerona

**W**- matrica težina izlaznih neurona

**BIAS**- neuron bez ulaza konstantnog izlaza jednakog jedan

**NRMS**- normalizirani korijen srednje kvadratne pogreške- mjera točnosti

## **Popis mjernih jedinica**

m [kg]- masa osobe

L [cm]- visina osobe

t [s]- vrijeme računanja

## 1. Uvod

Umjetne neuronske mreže su dio šireg područja zvanog umjetna inteligencija. Umjetna inteligencija se bavi problemima za koje čovjek pri rješavanju koristi inteligenciju, odnosno odgovarajuće misaone funkcije, a ne determinističke matematičke postupke. Općenito, ti problemi se odnose na procese asocijativnog zaključivanja, prepoznavanja, klasifikacije, planiranja, strukturiranja, podešavanja i slično. Ti problemi se najčešće temelje na nepotpunim ili neegzaktnim ulaznim informacijama. Pri čemu je glavni zadatak izbjeći računalni determinizam, oblikovanjem metoda koje se mogu prilagođavati problemima i evoluirati.

### 1.1. Umjetne neuronske mreže

Umjetne neuronske mreže su plod ideje da se pokuša modelirati biofiziologija ljudskog mozga, s ciljem razumijevanja i objašnjenja kako isti funkcionira [1]. Cilj ideje modeliranja biofiziologije ljudskog mozga je kreiranje modela sposobnog da procesira, odnosno prihvata, obrađuje, generira, pohranjuje i prenosi informacije, analogne aktivnostima ljudskog mozga [1]. William James 1890. godine postavlja tezu koja je poslužila za izgradnju osnovne strukture umjetnog neurona: „Aktivnost bilo koje točke mozga čovjeka predstavlja zbroj tendencija svih ostalih točaka da se „prazne“ odnosno ispaljuju u nju. Ove tendencije proporcionalne su:

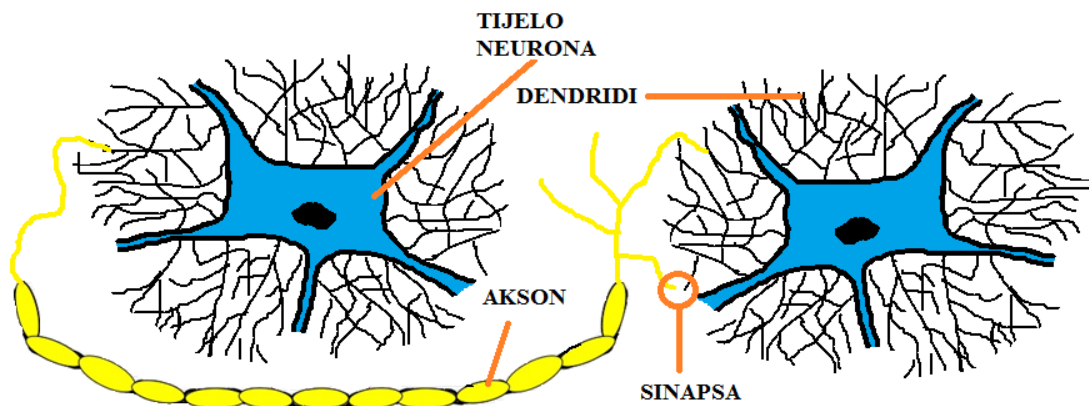
- broju točaka (broju veza) koje djeluju na promatranu točku,
- intenzitetu tih uzbuda (težini veza),
- odsutnosti rivalne točke koja nije u funkcionalnoj vezi s promatranom točkom, a u koju bi se „pražnjenje“ ostalih točaka moglo skrenuti.“[2]

1943. godine Warren McCulloch i Walter Pitts napravili su formalizaciju aktivnosti umjetnog neurona, osnovanoj na tvrdnji Williama Jamesa, predloživši jednostavan model umjetnog neurona. Takav umjetni neuron služi i danas kao osnova za izgradnju umjetnih neuronskih mreža.

Iako je sam model isti za sve umjetne neuronske mreže, one se međusobno razlikuju: po strukturi veza između neurona i samog neurona s okruženjem mreže te po načinu određivanja intenziteta tih veza (određivanja težina mreže), što predstavlja proces učenja mreže [1]. Iako se stvaranje umjetnih neuronskih mreža može podijeliti na dva po principu različita pristupa teško je odrediti točnu granicu između njih, jer se u oba pristupa koriste mnogi zajednički elementi. Jedan pristup ima za cilj realizirati modele koji dovoljno točno oponašaju aktivnosti ljudskog mozga, što bi trebalo rezultirati sustavom umjetne inteligencije. Drugi pristup podrazumijeva razvoj umjetnih neuronskih mreža s velikim matematičkim sposobnostima, kao što su neuroračunala koja su specificirana za rješavanje konkretnih zadataka praktične primjene. Peter J. W. Melsi termin „umjetne neuronske mreže“ koristi da bi se opisale različite topologije u visokom stupnju međusobno povezanih jednostavnih procesnih elemenata (neurona), koje nude alternativu (paralelno procesiranje) tradicionalnom pristupu računarstvu (serijska von Neumanova računala) [3]. Prednosti neuroračunala u odnosu na klasična računala su u paralelizmu pri procesiranju informacija, adaptibilnosti i sposobnosti učenja.

### 1.1.1. Biološki neuron

Pojednostavljeni biološki neuron sastoji se od: tijela, aksona i mnoštva dendrida koji okružuju tijelo neurona.



Slika 1. 1: Skica biološkog neurona

Tijelo neurona je glavni dio koji ima sve potrebne komponente u stanici: jezgru (koja sadrži DNK), endoplazmatska mrežica, ribosome (za izgradnju bjelančevina) i mitohondrije (za proizvodnju energije) [3]. Ako stanica tijela umre, umire neuron. Na tijelu neurona se nalazi akson i mnoštvo dendrida. Akson se može predložiti kao tanka cjevčica kojoj je jedan kraj povezan na tijelo neurona a drugi na niz grana. Krajevi ovih grana završavaju malim zadebljanjima koji najčešće dodiruju dendride, a rjeđe tijelo drugog neurona. Mali razmak između završetka aksona prethodnog neurona i dendrida ili tijela sljedećeg neurona naziva se sinapsa. Akson jednog neurona formira sinaptičke veze s mnoštvom drugih neurona.

Impulsi neurona putuju kroz akson do sinapsi odakle se signali različitog intenziteta šalju kroz dendride ili direktno na tijelo drugog neurona. Intenzitet impulsa je određen efikasnošću svakog pojedinačnog sinaptičkog prijenosa. Ukoliko je neuron doveden u stanje dovoljne uzbude poslat će impuls kroz svoj akson, realiziran njegovim trenutačnim stanjem te utjecajem signala od ostalih neurona preko njegovih dendrida ili njegovog tijela u kratkom vremenskom intervalu. Signali koje prenose dendridi mogu biti smirujući (sprječavajući, kočeći, inhibirajući) ili uzbudni [1]. Neuron će „ispaliti“, odnosno poslati, impuls kroz svoj akson ako je njegova uzbuda veća od smirujućeg utjecaja za kritični iznos, koji predstavlja prag osjetljivosti neurona [4].

Philip .D. Wasserman je procijenio da ljudski mozak ima preko 100 milijardi neurona te da neuroni komuniciraju kroz tijelo preko neuronskih vlakana, koji grade moguće 100 trilijuna veza. Ta mreža je odgovorna za sve procese i pojave u ljudskom tijelu kao što su: razmišljanje, svijest, emocije, autonomne funkcije poput disanja i rada srca, refleksi itd. Obzirom na količinu neurona i samih neuronskih veza može se zaključiti kolika je složenost same biološke neuronske mreže. Isto tako dodatnoj složenosti cijele biološke neuronske mreže pridonosi i činjenica da samo jedan neuron može generirati impuls koji će aktivirati ili smiriti na tisuće drugih neurona, iako ti isti neuroni mogu biti uzbuđivani od tisuća drugih neurona. Ovim se realizira visok stupanj povezanosti neurona u neuronskoj mreži, što upućuje na zaključak da se kompleksne funkcije neuronske mreže ostvaruju kompleksnošću veza među neuronima, a ne isključivo kompleksnošću svakog neurona ponaosob [2].

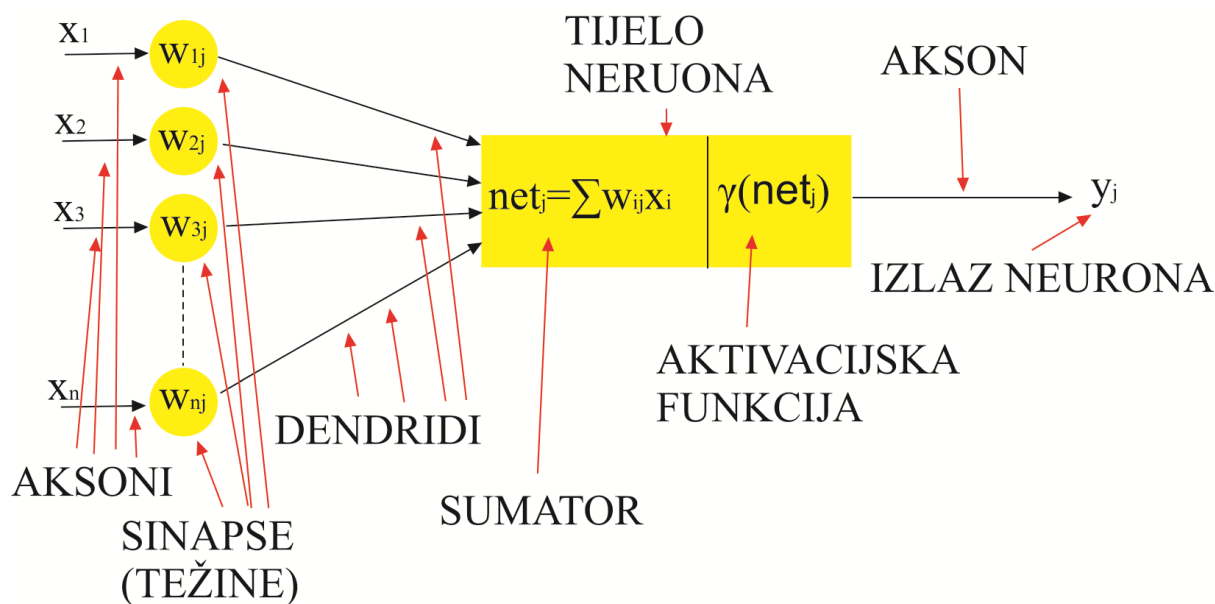
Iako je veliki dio funkcionalnih aktivnosti biološke neuronske mreže je već objašnjen, egzaktna načina rada biološke neuronske mreže nažalost nije još dovoljno razumljiv. Ovisno o obliku tijela veličina biološkog neurona je reda veličine 5 do 100 mikrona u promjeru te je identificirano nekoliko stotina bioloških neurona [4]. Ovisno o vrsti neurona, aksoni mogu biti prekriveni tankim slojem mijelinske ovojnice, kao što je izolator prevučen preko vodiča [6]. Različiti oblici tijela neurona pripisuju se specijalizaciji neurona za izvođenje određenih važnih funkcija. Komunikacija u mozgu se odvija na dva načina: kemijskim signalima preko sinapsi i električnim signalima unutar neurona [3]. Ovu izvanredno složenu aktivnost kreiranja obje vrste signala omogućava celularna membrana. Brzina ispaljivanja neurona je određena kumulativnim učinkom velikog broja uzbuđujućih i smirujućih ulaza, koji se obrađuju u tijelu neurona u kratkom vremenskom intervalu [1]. Uzbuđujući ulazi povećavaju, a smirujući smanjuju brzinu ispaljivanja neurona.

### **1.1.2. Sličnosti i razlike mozga i računala**

Računalo i ljudski mozak imaju nekoliko zajedničkih poveznica. Oba koriste električne signale, imaju veliki broj jednostavnih elemenata te izvode funkcije koje se općenito mogu nazvati računarskim funkcijama [1]. Međutim, postoje i više nego očite razlike među njima. Kako bi mozak riješio neki problem potrebno ga je učiti, dok je računalo potrebno programirati. Brzina prijenosa podataka je bitno različita, novija klasična računala tako imaju brzinu prijenosa reda veličine u nanosekundama, dok ljudski mozak prenosi podatke reda veličine u milisekundama [4]. Iako računala brže prenose podatke, u samom računanju je ljudski mozak neusporedivo brži. To je iz razloga što je ljudski mozak sačinjen od ogromnog broja paralelnih računarskih jedinica, odnosno neurona, koje omogućuju paralelno izvršavanje računarskih operacija. Kod klasičnih računala se sve odvija u serijama, odnosno tek nakon što je izračunata određena operacija može se računati sljedeća što znatno usporava sam proces izvođenja računanja. Nadalje, bitna razlika između računala i mozga je u tome da računalo neće praviti pogreške dokle god su svi ulazni podaci, program i sama oprema računala ispravni. Mozak, međutim, i za djelomično nekompletne i neispravne ulazne podatke daje dobre aproksimacije i zaključke. Mozak i kompjuter se mogu usporediti i u smislu potrošnje energije. Iako mozak teži samo 2% mase tijela troši preko 20% tjelesnog kisika te je mozak samim time i najkoncentriraniji potrošač energije u ljudskom tijelu [4]. Ukoliko se potrošnja mozga svede na snagu ispada da mozak troši samo 20W energije, što ga čini izuzetno uspješnim energetska članom. Za usporedbu, računalo koje bi imalo identičan kapacitet kao mozak čovjeka, trošio bi na tisuće puta energije [4].

### **1.1.3. Umjetni neuron**

Sam dizajn umjetnog neurona je osmišljen na način da oponaša osnovne funkcije biološkog neurona. Sumator umjetnog neurona zamjenjuje tijelo biološkog neurona dok ulogu dendrida preuzimaju ulazi u sumator, izlaz iz sumatora je akson umjetnog neurona. Prag osjetljivosti umjetnog neurona je izveden pomoću aktivacijske funkcije te ona simulira prag osjetljivosti biološkog neurona. Ostvarivanje veze umjetnog neurona sa njegovom okolinom postiže se pomoću težinskih faktora te oni preslikavaju funkcije sinaptičke veze biološkog neurona. Težinski faktor može biti različitih predznaka, a također može biti i neka funkcije kojom se postiže da je težinski faktor varijabilan.



Slika 1. 2: Umjetni neuron

Ukoliko je težinski faktor jednak nuli onda odgovarajuća veza s okolinom neurona ne postoji, pa se u shemi neuronske mreže ni ne ucrtava. Težinski faktori rade isto što i sinapse kod biološkog neurona: povezuju izlaze iz okoline neurona, odnosno izlaze drugih neurona s ulazima sumatora. Intenzitet te veze ovisi o iznosu, a karakter veze o predznaku težinskog faktora.

Izlaz sumatora povezuje se na ulaz aktivacijske funkcije, koja na izlazu daje izlaz umjetnog neurona. Aktivacijske funkcije mogu biti linearne i nelinearne. Kod linearnih aktivacijskih funkcija se izlaz sumatora množi sa nekim faktorom, odnosno pojačanjem te se tako dobiva izlaz neurona. Nelinearne aktivacijske funkcije mogu poprimiti različite oblike, ali najčešće se koriste: funkcije praga osjetljivosti, sigmoidalne, hiperbolične i harmoničke funkcije [5]. Nelinearne aktivacijske funkcije prevode izlaz sumatora na izlaz neurona preko nelinearnog pojačanja. Tako funkcija praga osjetljivosti daje na izlazu neurona jedinicu, ako je izlaz sumatora veći od zadanog praga osjetljivosti, što odgovara ispaljivanju impulsa kod biološkog neurona [1]. U suprotnom slučaju izlaz neurona je nula te je on neaktivan. Na primjer, ako se za aktivacijsku funkciju uzme signum funkcija, onda se svaki pozitivni bez obzira na iznos (beskonačno veliki ili da teži u nulu) izlaz sumatora će dati jedan, a svaki negativni bez obzira na iznos će dati minus jedan na izlazu. Na taj način umjetni neuron može uspješno obrađivati signale po iznosu i visokog i niskog intenziteta sa oba predznaka te je time riješen problem zasićenja. Iz svega navedenog se može zaključiti kako aktivnost umjetnog neurona ovisi o:

- broju ulaza iz okoline neurona,
- intenzitetu i karakteru uzbuđa,
- pragu osjetljivosti koji stanje neurona mora dosegnuti prije nego što „ispali impuls“ preko izlaza u okolinu neurona [9].

Najvažniji parametar ocjene sposobnosti umjetnih neuronskih mreža je kapacitet umjetne neuronske mreže a njega određuju broj neurona i broj slojeva neurona [9]. Neuronska mreža realizirana sa jednim jedinim neuronom, sa parametrom koji može poprimiti stotinu vrijednosti u različitim vremenskim trenucima, može imati veći kapacitet od



neuronske mreže sa desetak neurona i fiksnim parametrima [1]. Opisani umjetni neuron se često naziva prema svojim autorima McCulloch-Pittsiv neuron [2]. Ukoliko je aktivacijska funkcija oblika praga osjetljivosti takav neuron se u literaturi često naziva Perceptron. Jednostavna struktura opisanog umjetnog neurona zanemaruje mnoge karakteristike biološkog neurona. Tako, na primjer, ne vodi računa o kašnjenju signala (ulaz neurona producira trenutno njegov izlaz), što se odražava na dinamiku sustava, iako to u nekim slučajevima može biti prednost jer se povećava brzina procesiranja ulaznih signala mreže [3]. Jednako tako opisani umjetni neuron ne uključuje efekte sinkronizacije, odnosno frekventne modulacije biološkog neurona [5]. Sam umjetni neuron nije u mogućnosti realizirati funkcije, ali je to moguće postići odgovarajućim povezivanjem neurona u umjetnu neuronsku mrežu. Bez obzira na ograničenja opisanog umjetnog neurona, umjetne neuronske mreže realizirane povezivanjem takvih neurona mogu dati dobre rezultate.

#### **1.1.4. Podjela umjetnih neuronskih mreža**

Umjetne neuronske mreže se mogu ovisno o kriteriju podijeliti na nekoliko kategorija. Kategorije po kojima možemo podijeliti umjetne neuronske mreže su:

- prema broju slojeva,
- prema protoku signala: unaprijedne (FeedForward) i povratne (FeedBack),
- u odnosu na vrijeme: kontinuirane i diskretne i dr.[8].

Umjetne neuronske mreže prema broju slojeva se mogu podijeliti na: jednoslojne i višeslojne neuronske mreže. Paralelno složen skup neurona gradi jedan sloj neuronske mreže. Ukoliko se radi o višeslojnim mrežama uobičajeno je da te mreže imaju ulazni i izlazni sloj, a između njih se nalazi skriveni sloj.

Ukoliko se slojevi neuronske mreže povežu tako da signal putuje samo u jednom smjeru govori se o unaprijedim neuronskim mrežama, ako postoji barem jedna povratna petlja, kroz koju se signal vraća na ulaz, govori se o povratnim neuronskim mrežama. Prilikom rješavanja različitih problema bitno je koje od tih mreža će se koristiti, jer se neki problemi mogu riješiti unaprijednim neuronskim mrežama, dok se drugi problemi mogu riješiti pomoću povratnih neuronskih mreža. Također, postoje i mreže koje su kombinacija unaprijednih i povratnih mreža, tzv. Hibridi. Kod korištenja povratnih neuronskih mreža potrebno je naglasiti da uvođenjem povratne petlje u neuronsku mrežu uključuje se i vrijeme. Naime, kod povratnih neuronskih mreža izlaz nekog neurona može djelovati povratno na samog sebe i na druge neurone [1]. Iz toga je jasno da je nemoguće da se dogodi istovremeno, jer se najprije mora stvoriti izlaz neurona da bi se mogao povratno djelovati na samog sebe ili ostale neurone. Samim time operacije signala u sumatoru kao i signali postaju funkcije vremena [1].

#### **1.1.5. Učenje umjetnih neuronskih mreža**

U suštini učenje neuronskih mreža se razlikuje na supervizorno i nesupervizovrno učenje. Supervizorno učenje zahtjeva učitelja neuronske mreže, koji promatra ponašanje mreže te na temelju toga istu i ispravlja kako bi postigao željeno ponašanje. Prilikom korištenja ovog pristupa učenja umjetnih neuronskih mreža prvo je potrebno odrediti broj ulaza, broj slojeva, broj izlaza, broj težina mreže koji svi zajedno tvore strukturu neuronske mreže. Početne težine obično se usvajaju generiranjem slučajnih vrijednosti. Potom se na ulaz mreže dovodi skup ulaznih varijabli te mreža producira skup izlaznih varijabli. Izlazne varijable se potom uspoređuju sa željenim izlaznim varijablama. Razlika željenih i stvarnih

varijabli gradi pogrešku neuronske mreže, koja se koristi za računanje novih težina mreže preko određenog algoritma. Cijeli postupak se ponavlja iteracijski sve dok veličina pogreške ne postane manja od unaprijed određene veličine pogreške, ukoliko se radi o mrežama u kojima se pogreška računa iteracijskim postupkom. Međutim, postoje mreže koje uče u jednom koraku učenja te se nakon učenja tih mreža na izlazu ne pojavljuje pogreška učenja. Također, prilikom učenja je moguće po potrebi i mijenjati strukturu mreže. Završetkom procesa učenja slijedi proces testiranja neuronske mreže. Testiranje se provodi sa novim skupom ulaza koji nije sadržan u ulaznim podacima tijekom učenja. Prilikom testiranja mreža producira nove izlaze koji se uspoređuju za željenim izlazima, pritom se ne mijenjaju parametri mreže. Iznos pogreške u procesu testiranja služi za ocjenu robusnosti mreže, tj. sposobnosti mreže da daje zadovoljavajuće izlaze i za skup ulaza kojima nije bila učena.

Za razliku od supervizornog učenja, nesupervizorno učenje neuronske mreže ne koristi vanjskog učitelja, nego se mreža sama organizira. Mreže učene ovom metodom još se nazivaju i samoorganizirajuće neuronske mreže. Na ulaz mreže se dovodi skup ulaznih varijabli, a mreža po dobro definiranom algoritmu samoorganizira svoje parametre. Obzirom da željeni izlaz nije definiran za vrijeme učenja, rezultat učenja nije moguće predvidjeti. Tako se, na primjer, od neuronske mreže može zahtijevati da „slični“ skupovi ulaza aktiviraju izlaz jednog te istog neurona [1]. Tako organizirana mreža se može koristiti za klasifikaciju uzoraka, odnosno njihovo prepoznavanje. Kao i kod supervizornog učenja nakon procesa učenja se također provodi proces testiranja mreže. Postoji mnoštvo algoritama učenja neuronskih mreža neki od njih su: generalizirano delta pravilo, povratno propagiranje, optimalno učenje kod kojeg je potrebno zadati kriterij optimalnosti, adaptivno učenje i drugi.

## **1.2. Matlab**

Matlab je matematički programski paket za znanstveni i inženjerski numerički račun. Matlab je nastao kako bi omogućio studentima korištenje fortranskih paketa LINPACK i EISPACK bez potrebe za učenjem Fortran-a [10], a koji su prethodno desetljećima usavršavani. Prva verzija matlab-a je napravljena 1970-tih godina na sveučilištu University of New Mexico. Samo ime MATLAB dolazi od MATrix LABoratory, jer se na početku matlab temeljio na kompleksnoj matrici kao tipu podatka. Matlab ima odlične mogućnosti povezivanja sa programima pisanim u programskim jezicima C ili Fortran te ga time čini otvorenim i za složenije projekte. Matlab je jezik visoke učinkovitosti u računanju, ali i u vizualizaciji, simulacijama, programiranju u prozorskom okolišu. Danas Matlab višestruko nadilazi početni oblik matričnog laboratorija jer uz osnovni paket postoje i brojni paketi koji pokrivaju puno šira područja poput: obrade slike, obrade signala, automatskog upravljanja, identifikaciju sustava, statističke obrade, neuronskih mreža, 2D i 3D grafičko oblikovanje, financijske matematike i mnoge druge.

### **1.2.1. Programiranje u Matlab-u**

Programiranje u Matlab-u je moguće interaktivno pomoću naredbene linije. Međutim mana takvog unošenja je što zapis u naredbenoj liniji nije trajan te se stoga koriste M-datoteke u koje se unose željene naredbe koje je moguće spremiti. Slijed naredbi izvršavanja M-datoteke je odozdo prema gore, ukoliko nije drukčije definirano samim kodom programa. Svi podaci koji su unošeni tretiraju se kao matrice, isto tako se običan skalar tretira kao matrica 1x1. Obzirom na način unosa samih formula koji je vrlo blizak načinu na koji se inače zapisuju matematičke formule, jedan redak Matlab koda može zamijeniti na stotine redaka napisanih u nekom drugom programskom jeziku poput C, Pascala i sl. Velik broj naredbi koje

se koriste u Matlab-u su identične kao i u mnogim programima za programiranje što učenje programiranja u Matlab-u znatno olakšava.

### **1.2.2. Grafičko korisničko sučelje (eng. Graphic User Interface GUI)**

GUI je dio Matlab-a koji omogućava izradu grafičkih sučelja programiranjem odgovarajućih naredbi direktno u M-datoteku. Međutim takav način programiranja grafičkog sučelja bi bio izuzetno dugačak i kompliciran, u tu svrhu je napravljen GUIDE, odnosno alat za izradu grafičkog sučelja ubacivanjem gotovih objekata korisničkog sučelja.

### **1.2.3. GUIDE (Graphical User Interface development environment – Okolina za izradu grafičkog korisničkog sučelja)**

Pokretanja GUIDE sučelja se vrši preko glavnog matlab-ovog prozora te se otvara prozor GUIDE Layout Editor u koji se postavljaju željeni objekti te se time definira izgled budućeg grafičkog sučelja. Postavljanje objekata u površinu se radi jednostavnim „*drag and drop*“ postupkom. Neki od mogućih objekata su: statički tekst, okvir za unos teksta, tipke, pop-up izbornici i dr. Datoteka u koju se sprema izgled grafičkog sučelja je .fig prilikom samog spremanja .fig datoteke također nastaje i M-datoteka u koju se inicijaliziraju svi objekti te se u njoj nalaze sve informacije i funkcije koje izvode pojedini objekti.

Dvostrukim pritiskom na neki od objekata koji su postavljeni na površinu otvara se „Property Inspector“. U njemu se nalaze svi parametri koji se odnose na odabrani objekt, pritom je važno naglasiti da će različiti objekti imati i neke različite parametre. Neki od parametara su: oznaka (*tag*), povratni poziv (*callback*), vrijednost (*value*), boja i dr. Sve vrijednosti parametara se definiraju automatski te tako npr. oznaka odnosno *tag* za tipku poprima oznaku *pushbutton1* te se pod tim istim nazivom sprema i u M-datoteku. Preporučljivo bi bilo izmijeniti oznake u nazive koji bi bili bliži radnji koju izvršava ta tipka. Npr. tipka koja bi izvršavala zbrajanje dva broja bi se nazvala *zbroji*. Kako bi bilo moguće izvršavanje naredbe potrebno je u M-datoteci napraviti povratni poziv, odnosno *Callback* te unutar te funkcije definirati naredbu koja će biti izvršena prilikom pritiska te tipke. Nakon postavljanja svih željenih elemenata moguće je pokrenuti grafičko sučelje pritiskom na tipku *run*. Međutim, ukoliko u M-datoteku nisu upisane naredbe koje izvršavaju elementi u pokrenutom sučelju neće se ništa izvršavati.

Potrebno je naglasiti da je prilikom korištenja GUI-a potrebno sve vrijednosti koje se koriste proglasiti „stringovima“. String je niz znakova, a predstavljen je kao vektor ili jednodimenzionalna ili jednostupčana matrica.

## 2. Statička unaprijedna neuronska mreža s povratnim prostiranjem pogreške (eng. EBP- Error-Back Propagation)

Kako bi se objasnilo funkcioniranje statičke unaprijedne neuronske mreže s povratnim prostiranjem pogreške prvo će biti objašnjen osnovni dio neuronske mreže odnosno Perceptron, kako uči neuronska mreža, nakon toga će biti objašnjena matematika koja stoji iza učenja neuronske mreže.

### 2.1. Perceptron

Perceptron je umjetni neuron koji za aktivacijsku funkciju ima prag osjetljivosti. Aktivacijska funkcija koja se koristi je binarna funkcija praga osjetljivosti definirana izrazom:

$$y_i = f(z_i) = \begin{cases} 1 & \text{ako } z_i > \text{prag} \\ 0 & \text{ako je } z_i \leq \text{prag} \end{cases} \quad (2.1)$$

Dakle iz toga se može zaključiti da je Perceptron binarni neuron, koji na izlazu može dati samo 0 ili 1. Ukoliko se na izlazu nalazi 0 to znači da je promatrani Perceptron neaktivan, a ukoliko je na izlazu 1 znači da je aktivan te šalje signal neuronima s kojima je u doticaju. Ukoliko bi se više Perceptrona povezalo to bi nazivali perceptronska mreža. Ovisno kako su ti perceptroni povezani govorimo o više različitih vrsta perceptronskih mreža s obzirom na broj slojeva koji se u njoj nalaze. Jedan sloj neuronske mreže je paralelan skup neurona koji nisu međusobno povezani. Tako raspoznavamo: jednoslojne, dvoslojne i više slojne perceptronske mreže.

Struktura same perceptronske mreže je u potpunosti ovisna o klasifikacijskom problemu. Broj neurona ulaznog sloja je određen brojem različitih karakteristika, a broj izlaznih neurona je određen brojem jedinki za koje se provodi klasifikacija. Ulazni sloj služi za prijenos signala prema izlaznom sloju. Ukoliko se govori o višeslojnim perceptronskim mrežama prijenos signala sa ulaznog sloja ide preko sakrivenog ili sakrivenih slojeva do izlaznog sloja. Svi slojevi osim ulaznog sloja perceptronske mreže sadrže težine kojima se množi signal te se one sumiraju na ulasku u neuron. Težine svih slojeva su podesivi parametri te su one predmet učenja. Učenje se provodi iz skupa za učenje, koji se sastoji od ulaznih značajki i njihovih željenih izlaznih vrijednosti. Cilj učenja je podešavanje nepoznatih vrijednosti težina. Nakon uspješno završenog postupka učenja, perceptronske mreže mogu se koristiti kao klasifikatori.

#### 2.1.1. Učenje Perceptrona

Perceptronske mreže uče se iterativnim postupkom, po definiranom algoritmu:

1. Za ulazni vektor  $x$  uzorka iz skupa za učenje izračuna se izlazni vektor  $y$ :

$$y = xW, \quad (2.2)$$

gdje je  $W$  matrica težina, čiji se elementi u prvom koraku najčešće zadaju kao slučajni brojevi u intervalu  $(0,1)$

2. A) ukoliko je izlaz točan, povrat na točku 1,  
B) ukoliko je pogrešan, i ako je 0 dodaju se vrijednosti ulaznog vektora pripadajućim težinama,

C) ukoliko je pogrešan, i ako je 1 od pripadajućih težina se oduzmu vrijednosti ulaznog vektora,

3. Povrat na točku 1.

Jedan od prvih postupaka namještanja težina je Delta pravilo. Kao što je već navedeno  $W$  je jačina veze odnosno težina između dva neurona. Ukoliko je neuron  $i$  povezan sa neuron  $j$ ,  $w_{ij}$  označava težinu veze od neurona  $i$  prema neuronu  $j$ . ukoliko je neuron  $j$  također spojen i sa neuronima  $1, 2, \dots, n$  njihove težine su također pohranjene u varijablama  $w_{1j}, w_{2j}, \dots, w_{nj}$ . Promjena težina obavlja se prema ovom pravilu učenja:

$$W_{ij}^{novi} = \Delta W_{ij} + W_{ij}^{stari}, \quad (2.3)$$

gdje je  $\Delta W_{ij}$  vrijednost prilagođavanja težina veza od neurona  $i$  prema neuronu  $j$  izraženom prema:

$$\Delta W_{ij} = \eta \delta y_i, \quad (2.4)$$

gdje je  $y_i$  vrijednost izlaza izračunatog u neuronu  $i$ ,  $\eta$  je koeficijent brzine učenja, a  $\delta$  je greška izračunata prema:

$$\delta = T_j - A_j, \quad (2.5)$$

pri čemu  $T_j$  označava željenu izlaznu vrijednost izlaznog neurona  $j$ , a  $A_j$  označava trenutno odnosno izračunatu vrijednost izlaznog neurona  $j$ .

Postupak učenja iterativno se provodi za sve uzorke skupa za učenje tako dok svi nisu pravilno razvrstani u pripadajuće im skupine. Dokazano je da postupak učenja uspješno konvergira u konačnom broju koraka. Međutim, studija koju su proveli Minsk i Paperta dokazala je da postoji velika skupina vrlo jednostavnih problema koje nije moguće riješiti jednoslojnom perceptronskom mrežom. Glavni uzroci te nemogućnosti su u tome što su to linearno neseeparibilni uzorci. Najpoznatiji od tih problema je ekskluzivni ILI problem (eng. XOR). Međutim, taj problem se riješio dodavanjem dodatnih (sakrivenih) slojeva ali uz uvjet da se u tim slojevima koriste linearne aktivacijske funkcije. Zadatak dodatnih slojeva neurona je da razdvoje linearno neseeparibilne uzorke. Dodatan problem je predstavljala činjenica što nije postojao učinkovit postupak učenja težina skrivenih slojeva, budući da se delta pravilo nije moglo primijeniti na neurone skrivenih slojeva.

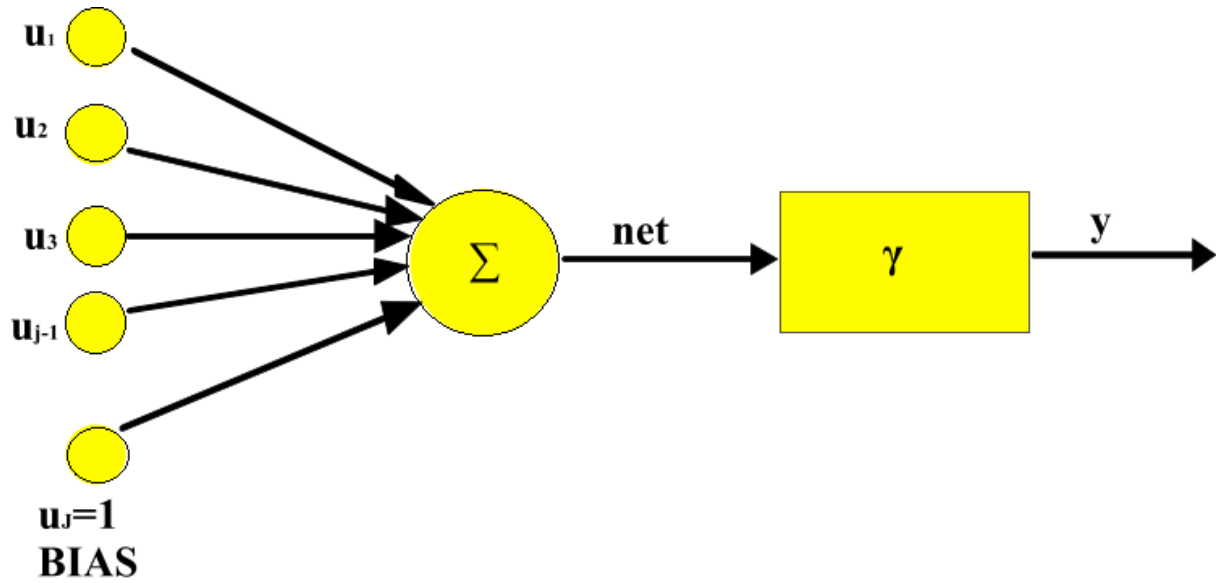
## 2.2. Višeslojne statičke neuronske mreže

Višeslojne neuronske mreže najpoznatiji i najčešće upotrebljavani oblik neuronskih mreža. Kao što je već rečeno ranije služe za klasifikaciju uzoraka. Osnovni dio od kojih je sastavljena višeslojna statička neuronska mreža je statički neuron.

### 2.2.1. Model statičkog neurona

Model statičkog neurona je sastavljen dvije temeljne pod funkcije, to su funkcija sume i aktivacijska funkcija. Bitno je naglasiti da je osnovna značajka statičkog neurona posjedovanje više ulaza i samo jedan izlaz. Kako bi se učenje moglo uspješno odvijati potrebno je da svaki neuron koji sudjeluje u procesu učenja posjeduje poseban ulaz jedinične

vrijednosti koji je u strukturi neuronske mreže realiziran vezom sa zasebnim neuronom oznake Bias, konstantnog izlaza jednakog jedinici [1].



Slika 2. 1: Model statičkog neurona

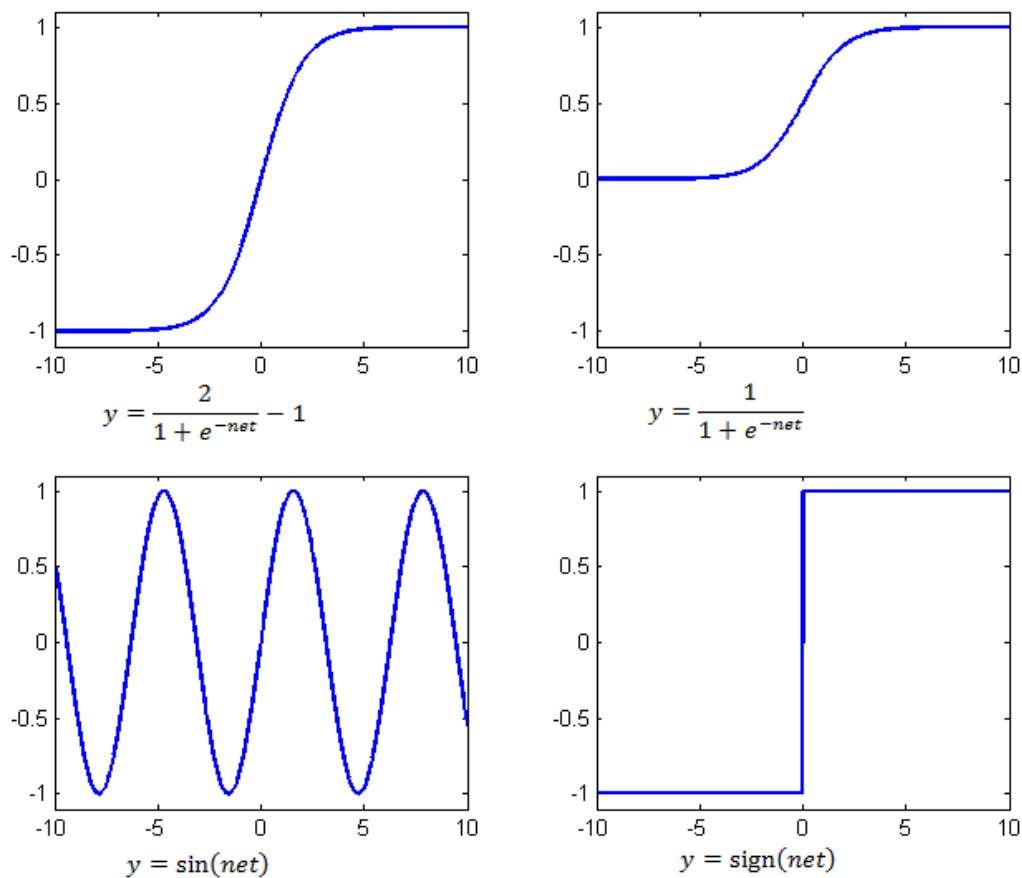
Funkcija sume statičkog neurona jednaka je sumi umnožaka izlaza neurona i njihovih pripadajućih težina, a rezultira vrijednošću  $net$  [1].

$$net = \sum_{j=1}^J \mathbf{w}_j u_j. \quad (2.6)$$

Aktivacijska funkcija  $\gamma$  izvršava preslikavanje vrijednosti  $net$  u izlaznu vrijednost neurona  $y$  [1]:

$$y = \gamma(net). \quad (2.7)$$

Za aktivacijsku funkciju se najčešće se koriste monotono rastuće funkcije sa zasićenjem. Neke od aktivacijskih funkcija su prikazane slikom.



Slika 2. 2: Prikaz nekih aktivacijskih funkcija

Najčešće korištena aktivacijska funkcija neurona je nelinearna bipolarna sigmoidalna funkcija prikazana na slici u prvom dijagramu. Ta aktivacijska funkcija je globalnog karaktera i pokazuje dobre rezultate i kad se koristi u učenju dinamičkih neuronskih mreža [1]. U ovom radu će biti prikazano učenje pomoću dvije aktivacijske funkcije: nelinearna bipolarna sigmoidalna funkcija i obična sinusna funkcija. Ukoliko odaberemo nelinearnu bipolarnu sigmoidalnu funkcija vrijednost  $y$  poprima sljedeći oblik:

$$y = \frac{2}{1 + e^{-net}} - 1, \quad (2.8)$$

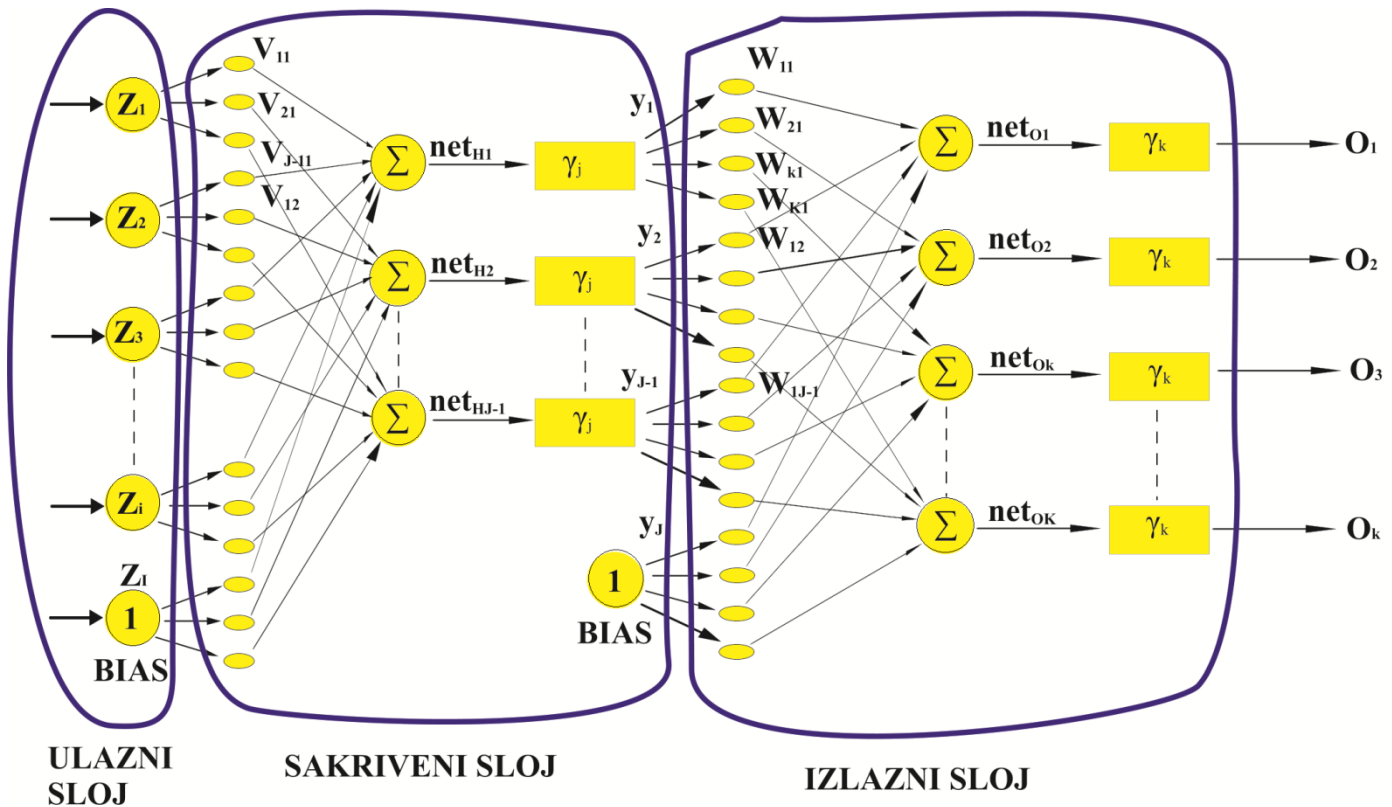
a ukoliko odaberemo za aktivacijsku funkciju običnu sinusnu  $y$  poprima sljedeći oblik:

$$y = \sin(net). \quad (2.9)$$

### 2.2.2. Model statičke neuronske mreže

Model statičke neuronske mreže je sastavljen od niza statičkih neurona koji su raspodijeljeni u slojeve. Slojevi mreže su međusobno povezani težinskim faktorima odnosno

težinama neurona. Slojeve mreže dijelimo na tri vrste: ulazni, sakriveni i izlazni sloj. Ulazni i izlazni sloj su u direktnoj vezi sa okolinom, dok skriveni sloj to nije, pa odakle i naziv „skriveni“ [1].



Slika 2. 3: Model statičke unaprijedne neuronske mreže

Na slici 2.3 je prikazan model statičke unaprijedne neuronske mreže. Kao što je vidljivo iz slike podaci koji ulaze u mrežu dolaze na ulazni sloj, koji se označava sa  $Z_i$ . Uobičajeno je da neuroni u ulaznom sloju nisu neuroni u pravom smislu, nego su oni u svojoj biti čvorovi za distribuciju ulaza neuronske mreže prema prvom sakrivenom sloju neurona. Iz slike je također vidljivo da je ulazni sloj povezan sa sakrivenim slojem opterećenim težinama  $v_{jl}$ . Također, iz slike je vidljivo kako su gotovi svi neuroni u potpunosti umreženi, odnosno da je svaki neuron pojedinog sloja povezan sa svakim neuronom prethodnog sloja. Iznimku čine neuroni označeni sa BIAS koji se nalaze u sakrivenom sloju te imaju konstantnu vrijednost jedan. Broj sakrivenih slojeva neurona je proizvoljan, ali najčešće se koriste jedan ili dva sloja. Teorijski radovi istraživača kao što su Cybenko i Funahashi pokazuju da je jedan jedini skriveni sloj neurona dostatan za dobru aproksimaciju bilo koje kontinuirane funkcije, uz uvjet da takav sloj ima dovoljno neurona [1]. Međutim, koliko je neurona potrebno ovisi o samom zadatku te je potrebno odrediti ih eksperimentalno. Što se tiče izlaznog sloja, on je povezan sa sakrivenim slojem neurona težinskim vrijednostima  $w_{jk}$  te on definiran izlaze neuronske mreže označene sa  $O_k$ .

### 2.2.3. Učenje povratnim rasprostranjem pogreške

Učenje povratnim rasprostranjem pogreške je način učenja s učiteljem. Sam postupak učenja je podešavanje težinskih koeficijenata veza između svih slojeva mreža sa ciljem da se



dobiveni izlazi mreže što više približe željenim izlazima, za odgovarajuće skupove ulaza. S obzirom da je cilj da se mreža što više približi željenim vrijednostima lako se zaključuje da postupak učenja ne daje sto postotnu točnost. Izlaz mreže uvijek se aproksimacija željenih izlaza. Sama kvaliteta aproksimacije željenih izlaza ovisi o nekoliko čimbenika, kao što su: sam zadatak učenja, odabranoj topologiji mreže te o odabranom algoritmu učenja. Algoritmi učenja se mogu podijeliti u dvije osnovne skupine, algoritme jednog koraka učenja i iteracijske algoritme učenja. Osnovna karakteristika učenja s učiteljem je da se u iterativnom postupku učenja, tj. iterativnom postupku adaptacije težinskih koeficijenata veza, mreži uzastopce prikazuju ulazne veličine i za njih odgovarajuće, željene izlazne veličine [1].

Postupak učenja se za svaki korak učenja odvija u dvije faze: unaprijedna i povratna faza. Sama promjena parametara se može također odvijati na dva načina: po skupu (eng. batch) ili po uzorku (eng. pattern). Ukoliko se učenje tj. promjena parametara odvija po skupu nakon svakog ulazno-izlaznog para vrijednost pogreške se sprema i dodaje prethodnoj pogrešci u tom koraku učenja. Nakon što prođu svi ulazno-izlazni parovi i sve vrijednosti grešaka se zbroje potom se mijenjaju parametri učenja. Tim načinom učenja promjena parametra se odvija nekom srednjom pogreškom učenja. Za razliku od učenja po skupu, učenje po uzorku mijenja parametre učenja nakon svakog ulazno-izlaznog para. Odnosno, nakon što prođe prvi ulazno-izlazni par izračuna se greška te se odmah pristupa mijenjaju parametara po definiranom algoritmu učenja.

## 2.2.4. Unaprijedna faza učenja

U unaprijednoj fazi učenja iz datoteke ulaznih veličina uzimaju se vrijednosti svih ulaza mreže  $Z$  te se na temelju tih ulaza izračunava izlaz mreže  $O$ . Kako bi to bilo moguće potrebno je prvo odrediti iznose težinskih koeficijenata  $V$  i  $W$ . Najčešće se te vrijednosti određuju generiranjem slučajnih vrijednosti brojeva. Raspon u kojem se odabiru slučajne vrijednosti brojeva je preporučen u istom redu veličine u kojem se nalaze ulazi i izlazi mreže, odnosno ukoliko su ulazi i izlazi normirani na vrijednosti između  $-1$  i  $1$ , preporučljivo je odabrati slučajne vrijednosti  $V$  i  $W$  također u tom rasponu. Međutim, ukoliko u samom početku učenja pojave problemi sa lokalnim minimumima pogreške, potrebno je interval početnih vrijednosti smanjiti za jedan red u odnosu na interval ulaza i izlaza, znači na  $-0.1$  i  $0.1$ .

Unaprijedna faza počinje funkcijom sume net neurona sakrivenog sloja  $H$ , dobiva se prvi indeks oznake sloja ( $netH$ ) te za svaki  $j$ -ti neuron dobiva drugi indeks te se računa na sljedeći način:

$$net_{Hj} = \sum_{i=1}^I v_{ji} Z_i, \quad j=1,2,\dots,J-1, \quad i=1,2,\dots,I, \quad (2.10)$$

gdje je  $I$  broj ulaznih neurona plus jedan (Bias), a  $J$  predstavlja broj neurona u sakrivenom sloju uvećan za jedan (Bias). Ukoliko za aktivacijsku funkciju sakrivenog sloja odaberemo nelinearnu bipolarnu sigmoidalnu funkciju onda se izlazi neurona mogu izračunati:

$$y_j = \frac{2}{1 + e^{-net_{Hj}}} - 1, \quad (2.11)$$

a ukoliko se za aktivacijsku funkciju odabere sinusna funkcija izlazi neurona se izračunavaju:

$$y = \sin(net_{Hj}) \quad (2.12)$$

gdje je  $j=1,2,\dots,J-1$ .

Izračunate vrijednosti izlaza neurona sakrivenog sloja su povezane sa ulazima neurona izlaznog sloja preko težinskih koeficijenata  $w_{kj}$ .

Funkcija sume  $net$  neurona izlaznog sloja  $O$  poprima prvi indeks pripadnog sloja ( $net_O$ ), a drugi pripadni indeks svakom neuronu izlaznog sloja, a izračunava se izrazom:

$$net_{Ok} = \sum_{j=1}^J w_{kj} y_j, \quad k = 1, 2, \dots, K, \quad (2.13)$$

gdje je  $K$  broj neurona izlaznog sloja, odnosno broj izlaza mreže.

Za aktivacijsku funkciju u ovom radu se koristi linearna aktivacijska funkcija pojačanja  $K_p=1$  te iz toga slijedi:

$$O_k = K_p net_{Ok}, \quad k = 1, 2, \dots, K. \quad (2.14)$$

Iako je u ovom radu korištena linearna aktivacijska funkcija, moguće je također i koristiti neke druge funkcije kao npr. sigmoidalnu aktivacijsku funkciju. Međutim, ukoliko se koristi funkcija sa zasićenjem na izlazu nije moguće ostvariti izlaze iz neuronske mreže veće od 1 i manje od -1.

## 2.2.5. Povratna faza učenja

U nastavku će biti objašnjene obje vrste povratnih faza, prvo po uzorku, a potom po skupu.

### 2.2.5.1. Povratna faza po uzorku

Nakon što je završena unaprijedna faza, dobiveni izlaz mreže se oduzima od željenog izlaza te se na temelju rezultata izračunava pogreška učenja. Na temelju pogreške učenja vrši se korekcija vrijednosti težinskih koeficijenata veza između slojeva. Postupak se ponavlja za svaki ulazno-izlazni par podataka učenje te se postupak prekida kad greška učenja bude jednaka ili manja od dozvoljene pogreške koju određuje učitelj.

Uobičajena statistička metoda regresijske analize, suma kvadrata pogreške kao mjera odstupanja izlaza mreže od željene vrijednosti izlaza najčešće je korištena funkcija cilja :

$$E = \frac{1}{2} \sum_{n=1}^N (d_n - O_n)^2, \quad (2.15)$$

gdje je  $N$  broj elemenata u skupu za učenje, odnosno broj podataka učenja iz zapisnika učenja ( $1/2$  pojednostavljuje potrebite derivacije funkcije cilja).  $O_n$  je dobiveni izlaz mreže, a  $d_n$  željeni izlaz mreže. Iz gornje jednadžbe slijedi da je postupak podešavanja težinskih koeficijenata minimizira funkciju cilja. Odabranom funkcijom cilja se vrši promjena koeficijenata težina primjenom nekog od algoritma nelinearnog optimiranja. Forma promjene parametara učenja težinskih koeficijenata  $\theta$ :

$$\mathcal{G}(n+1) = \mathcal{G}(n) + \Delta\mathcal{G}(n), \quad (2.16)$$

gdje je  $n$  trenutni korak učenja,  $\mathcal{G}=v$  za sakriveni sloj te  $\mathcal{G}=w$  za izlazni sloj,  $\Delta\mathcal{G}(n)$  veličina promjene parametara učenja,  $\mathcal{G}(n+1)$  je nova vrijednost parametara učenja. Pogrešku  $E(\mathcal{G})$  moguće je u okolišu točke  $\mathcal{G}$  aproksimirati s prva dva člana Taylorovog reda [1]:

$$E(\mathcal{G} + \Delta\mathcal{G}) \approx E(\mathcal{G}) + \Delta E(\mathcal{G}), \quad (2.17)$$

$$\Delta E(\mathcal{G}) = \Delta\mathcal{G}^T \nabla E(\mathcal{G}), \quad (2.18)$$

$$\nabla E(\mathcal{G}) = \frac{\partial E(\mathcal{G})}{\partial(\mathcal{G})}. \quad (2.19)$$

Izraz (2.19) naziva se gradijentom pogreške. Kako bi se pogreška smanjivala maksimalnim mogućim iznosom, potrebno je odrediti  $\Delta\mathcal{G}$  za koji pogreška učenja  $\Delta E(\mathcal{G})$  poprima najveći negativni iznos, a to se ostvaruje uz uvjet:

$$\Delta\mathcal{G} = -\eta \nabla E(\mathcal{G}), \quad (2.20)$$

gdje je  $\eta$  koeficijent brzine učenja, odnosno mjera te promjene. Koeficijent brzine učenja određuje učitelj, a iznos je najčešće između  $10^{-3}$  do 10. Izraz (2.20) osigurava promjenu težinskih koeficijenata u smjeru najstrmijeg pada ukupne pogreške. Uvrštavanjem izraza 20 u izraz 16 dobiva se:

$$\mathcal{G}(n+1) = \mathcal{G}(n) - \eta \nabla E(\mathcal{G}(n)). \quad (2.21)$$

Algoritam dan izrazom (2.21) poznat je u literaturi o nelinearnom optimiranju pod nazivom algoritam najstrmijeg pada, dok je prema literaturi o neuronskim mrežama poznat pod nazivom algoritam povratnog prostiranja pogreške i predstavlja najpoznatiji i najčešće primjenjivani način promjene parametara učenja [1].

### 2.2.5.2. Povratna faza po skupu

Kao što je već objašnjeno povratna faza po skupu se odvija na način da se sve greške ulazno-izlaznih parova zbrajaju te da se nakon toga pristupa promjeni parametara učenja. Tako da izraz (2.20) prelazi u:

$$\Delta\mathcal{G} = -\eta \sum_{i=1}^n \nabla E(\mathcal{G}), \quad (2.22)$$

gdje je  $n$  jednak broju ulazno-izlaznih parova. Iz toga slijedi da izraz (2.21) postaje:

$$\mathcal{G}(n+1) = \mathcal{G}(n+1) = \mathcal{G}(n) - \eta \sum_{i=1}^n \nabla E(\mathcal{G}). \quad (2.23)$$

### 2.2.6. Ubrzanje iterativnog učenja

Poznata je činjenica da je najveći nedostatak algoritma povratnog procesiranja pogreške veliki broj potrebnih iteracija. Kako bi se smanjila količina potrebnih iteracija, odnosno kako

bi se ubrzalo učenje i smanjio broj učenja koriste se različite modifikacije osnovnog algoritma. Jedan od mogućih izraza za ubrzanje promjene parametara učenja je takozvani zamah, odnosno momentum.

### 2.2.6.1. Zamah prvog reda

U jednadžbu za algoritam povratnog prostiranja pogreške se dodaje dio u kojem se koristi pogreška učenja u prethodnom koraku  $\Delta \mathcal{G}(n-1)$  pomnožena za koeficijentom zamaha  $\alpha$ . Sam iznos koeficijenta zamaha određuje učitelj te se on najčešće postavlja između 0.1 i 0.9. dodavanjem izraza za zamah jednadžba (2.21) poprima sljedeći oblik:

$$\mathcal{G}(n+1) = \mathcal{G}(n) - \eta \nabla E(\mathcal{G}(n)) + \alpha \Delta \mathcal{G}(n-1), \quad (2.24)$$

ako se radi o povratnoj fazi po uzorku. Ukoliko se radi o povratnoj fazi po skupu izraz (2.23) poprima sljedeći oblik:

$$\mathcal{G}(n+1) = \mathcal{G}(n) - \eta \sum_{i=1}^n \nabla E(\mathcal{G}) + \alpha \Delta \mathcal{G}(n-1). \quad (2.25)$$

### 2.2.6.2. Zamah drugog reda

Kako bi se dodatno ubrzao proces učenja moguće je dodati i zamah drugog reda. Međutim, nije uvijek moguće ostvariti ubrzanje pa ponekad zamah drugog reda može i usporiti samu brzinu učenja. Zamah drugog reda se oduzima od jednadžbe za algoritam povratnog prostiranja pogreške umnoškom koeficijenta zamaha drugog reda  $\beta$  i greške učenje u predzadnjem koraku  $\Delta \mathcal{G}(n-2)$ . Koeficijent zamaha drugog reda se izračunava preko sljedećeg izraza:

$$\beta = \frac{\alpha - 1}{3}. \quad (2.26)$$

Nakon ubacivanja izraza (2.26) u izraz (2.24) dobiva se novi izraz za algoritam povratnog prostiranja pogreške:

$$\mathcal{G}(n+1) = \mathcal{G}(n) - \eta \nabla E(\mathcal{G}(n)) + \alpha \Delta \mathcal{G}(n-1) - \beta \Delta \mathcal{G}(n-2), \quad (2.27)$$

ukoliko se radi o povratnoj fazi po uzorku. Ako se radi o povratnoj fazi po skupu izraz (2.25) poprima sljedeći oblik:

$$\mathcal{G}(n+1) = \mathcal{G}(n) - \eta \sum_{i=1}^n \nabla E(\mathcal{G}) + \alpha \Delta \mathcal{G}(n-1) - \beta \Delta \mathcal{G}(n-2). \quad (2.28)$$

## 2.2.7. Promjena težina izlaznog sloja

Promjena parametara učenje prilikom korištenja povratne faze povratnim prostiranjem pogreške se odvija od izlaznog sloja prema unutarnjem sloju mreže. Promjena težinskih faktora veza između i sakrivenog sloja odvija se prema izrazu (2.27) za povratno djelovanje po uzorku prema izlazu:

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) - \eta \nabla E(n) + \alpha \Delta \mathbf{w}_{kj}(n-1) - \beta \Delta \mathbf{w}_{kj}(n-2). \quad (2.29)$$

Odnosno, ako se koristi povratno djelovanje po skupu odvija se po izrazu (2.28) po izrazu:

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) + \eta \sum_{i=1}^n \nabla E(n) + \alpha \Delta \mathbf{w}_{kj}(n-1) - \beta \Delta \mathbf{w}_{kj}(n-2). \quad (2.30)$$

Kako bi bilo moguće pristupiti promjeni težina potrebno je prvo naći gradijent pogreške za težine izlaznog sloja odnosno  $w_{kj}$  koji se izračunava prema izrazu:

$$\nabla E(n) = \frac{\partial E(n)}{\partial \mathbf{w}_{kj}}, \quad (2.31)$$

gdje je  $n$  broj ulazno-izlaznog para. Prema gorenjem izrazu osnovni zadatak postupka učenja je određivanje pripadajućeg gradijenta pogreške. Izraz (2.31) se jednostavno rješava primjenom uzastopnih parcijalnih derivacija kako prikazuje slika 3:

$$\frac{\partial E(n)}{\partial w_{kj}} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{Ok}} \frac{\partial net_{Ok}}{\partial w_{kj}}. \quad (2.32)$$

Iz gornjeg izraza je lako izračunati pojedine članove, izraz (2.15) nakon deriviranja pojedinog člana daje:

$$\frac{\partial E(n)}{\partial O_k} = -(d_k - O_k). \quad (2.33)$$

Iz izraza (2.14) slijedi:

$$\frac{\partial O_k}{\partial net_{Ok}} = \dot{\gamma} = 1. \quad (2.34)$$

Množenjem izraza (2.33) i (2.34) pokazuje se karakteristična vrijednost algoritma povratnog prostiranja pogreške koja se prvo računa za izlazni sloj  $i$  koja je po definiciji:

$$\delta = -\frac{\partial E(n)}{\partial net}, \quad (2.35)$$

$$\delta_{Ok} = d_k - O_k. \quad (2.36)$$

Izraz (2.13) nakon deriviranja daje:

$$\frac{\partial net_{Ok}}{\partial \mathbf{w}_{kj}} = y_j. \quad (2.37)$$

Uvrštavanjem izraza (2.33), (2.34) i (2.37) u izraz (2.29) ukoliko se koristi povratna faza po uzorku dobiva se:

$$\nabla E(n) = \frac{\partial E(n)}{\partial \mathbf{w}_{kj}} = -(d_k - O_k) y_j = -\delta_{Ok} y_j. \quad (2.38)$$

Međutim, ukoliko se koristi povratna faza po skupu potrebno je prvo zbrojiti sve gradijente pogreške te nakon toga provesti učenje.

Uvrštavanjem izraza (2.38) u izraz (2.29) dobivamo konačni algoritam za izračunavanje promjene težina izlaznog sloja u slučaju korištenja povratne faze po uzorku:

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) + \eta \delta_{ok} y_j + \alpha \Delta \mathbf{w}_{kj}(n-1) - \beta \Delta \mathbf{w}_{kj}(n-2). \quad (2.39)$$

Ukoliko se koristi povratna faza učenje po skupu izraz (2.28) nakon uvrštavanja izraza (2.38) poprima sljedeći oblik:

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) + \eta \sum_{i=1}^n \delta_{ok} y_j + \alpha \Delta \mathbf{w}_{kj}(n-1) - \beta \Delta \mathbf{w}_{kj}(n-2). \quad (2.40)$$

## 2.2.8. Promjena težina sakrivenog sloja

Nakon što je izvršena promjena težina izlaznog sloja potrebno je promijeniti težine sakrivenog sloja. Temeljna jednadžba prema kojoj se vrši adaptacija težina sakrivenog sloja u svojoj se biti ne razlikuje od jednakosti (2.29) ukoliko se koristi povratna faza po uzorku [1]:

$$\mathbf{v}_{ji}(n+1) = \mathbf{v}_{ji}(n) - \eta \nabla E(n) + \alpha \Delta \mathbf{v}_{ji}(n-1) - \beta \Delta \mathbf{v}_{ji}(n-2). \quad (2.41)$$

Jednako tako temeljna jednadžba adaptacije težina sakrivenog sloja, prilikom korištenja povratne faze po skupu, se ne razlikuje od jednakosti (2.30):

$$\mathbf{v}_{ji}(n+1) = \mathbf{v}_{ji}(n) + \eta \sum_{i=1}^n \nabla E(n) + \alpha \Delta \mathbf{v}_{ji}(n-1) - \beta \Delta \mathbf{v}_{ji}(n-2). \quad (2.42)$$

Kao i za težine izlaznog sloja, tako i za težine ulaznog sloja jedini problem je izračunati gradijent pogreške  $\nabla E(n)$  [1]. Kao i kod izračunavanja gradijenta pogreške za izlazni sloj, primjenom uzastopnih parcijalnih derivacija, isti postupak se primjenjuje za izračunavanje gradijenta pogreške za sakriveni sloj kako prikazuje slika 2.3:

$$\frac{\partial E(n)}{\partial \mathbf{v}_{ji}} = \frac{\partial E(n)}{\partial y_j} \frac{\partial y_j}{\partial \text{net}_{Hj}} \frac{\partial \text{net}_{Hj}}{\partial \mathbf{v}_{ji}}. \quad (2.43)$$

Iz slike 2. 3 je vidljivo da na promjenu svake težine sakrivenog sloja utječu svi neuroni izlaznog sloja. Iz toga slijedi kako prvi razlomak u izrazu (2.43) poprima sljedeći oblik:

$$\frac{\partial E(n)}{\partial y_j} = \frac{\partial E(n)}{\partial O_1} \frac{\partial O_1}{\partial \text{net}_{o1}} \frac{\partial \text{net}_{o1}}{y_j} + \frac{\partial E(n)}{\partial O_2} \frac{\partial O_2}{\partial \text{net}_{o2}} \frac{\partial \text{net}_{o2}}{y_j} + \dots + \frac{\partial E(n)}{\partial O_K} \frac{\partial O_K}{\partial \text{net}_{oK}} \frac{\partial \text{net}_{oK}}{y_j}, \quad (2.44)$$

gdje je:

$$\frac{\partial E(n)}{\partial O_k} = -(d_k - O_k), \quad k = 1, 2, \dots, K, \quad (2.45)$$

$$\frac{\partial O_k}{\partial net_{Ok}} = 1, \quad k = 1, 2, \dots, K, \quad (2.46)$$

$$\frac{\partial net_{Ok}}{\partial y_j} = \mathbf{w}_{kj}, \quad k = 1, 2, \dots, K \text{ i } j = 1, 2, J-1. \quad (2.47)$$

Uvrštavanjem izraza (2.45), (2.46) i (2.47) u izraz (2.44) dobiva se:

$$\frac{\partial E(n)}{\partial y_j} = - \sum_{k=1}^K (d_k - O_k) \mathbf{w}_{kj}, \quad (2.48)$$

uvrštavanjem izraza (2.36) u izraz (2.48) prvi razlomak izraza (2.43) poprima konačni oblik:

$$\frac{\partial E(n)}{\partial y_j} = - \sum_{k=1}^K \delta_{Ok} \mathbf{w}_{kj}. \quad (2.49)$$

Drugi razlomak sa desne strane jednakosti izraza (2.43) određuje se iz izraza (2.11). Ukoliko se koristi nelinearna bipolarna sigmoidalna aktivacijska funkcija poprima oblik:

$$\frac{\partial y_j}{\partial net_{Hj}} = \dot{y}_j = \frac{2e^{-net_{Hj}}}{\left(1 + e^{-net_{Hj}}\right)^2}, \quad (2.50)$$

Ukoliko iz izraza (2.11) izlučimo  $e^{-net_{Hj}}$  izraz (2.50) poprima sljedeći oblik:

$$\frac{\partial y_j}{\partial net_{Hj}} = \dot{y}_j = \frac{1}{2} (1 - y_j^2). \quad (2.51)$$

Ukoliko se koristi izraz (2.12), odnosno sinusna aktivacijska funkcija, drugi razlomak sa desne strane izraza (2.43) poprima oblik:

$$\frac{\partial y_j}{\partial net_{Hj}} = \dot{y}_j = \cos(net_{Hj}). \quad (2.52)$$

Treći razlomak s desne strane jednakosti izraza (2.43) određuje se iz izraza (2.10) te poprima oblik:

$$\frac{\partial net_{Hj}}{\partial v_{ji}} = Z_i. \quad (2.53)$$

Ukoliko se koristi nelinearna bipolarna sigmoidalna aktivacijska funkcija i povratno prostiranje pogreške po uzorku izraz (2.41) poprima sljedeći oblik:

$$\mathbf{v}_{ji}(n+1) = \mathbf{v}_{ji}(n) + \frac{1}{2} \eta (1 - y_j^2) Z_i \sum_{k=1}^K \delta_{Ok} \mathbf{w}_{kj} + \alpha \Delta \mathbf{v}_{ji}(n-1) - \beta \Delta \mathbf{v}_{ji}(n-2). \quad (2.54)$$

Ukoliko se koristi sinusna aktivacijska funkcija i povratno prostiranje pogreške po uzorku izraz (2.41) poprima sljedeći oblik:

$$\mathbf{v}_{ji}(n+1) = \mathbf{v}_{ji}(n) + \eta \sin(\text{net}_{Hj}) Z_i \sum_{k=1}^K \delta_{Ok} \mathbf{w}_{kj} + \alpha \Delta \mathbf{v}_{ji}(n-1) - \beta \Delta \mathbf{v}_{ji}(n-2). \quad (2.55)$$

Nadalje, ukoliko se koristi nelinearna bipolarna sigmoidalna aktivacijska funkcija i povratno prostiranje pogreške po skupu izraz (2.42) poprima sljedeći oblik:

$$\mathbf{v}_{ji}(n+1) = \mathbf{v}_{ji}(n) + \eta \sum_{i=1}^n \frac{1}{2} (1 - y_j^2) Z_i \sum_{k=1}^K \delta_{Ok} \mathbf{w}_{kj} + \alpha \Delta \mathbf{v}_{ji}(n-1) - \beta \Delta \mathbf{v}_{ji}(n-2). \quad (2.56)$$

Te ukoliko se koristi sinusna aktivacijska funkcija i povratno prostiranje pogreške po skupu izraz (2.42) poprima sljedeći oblik:

$$\mathbf{v}_{ji}(n+1) = \mathbf{v}_{ji}(n) + \eta \sum_{i=1}^n \sin(\text{net}_{Hj}) Z_i \sum_{k=1}^K \delta_{Ok} \mathbf{w}_{kj} + \alpha \Delta \mathbf{v}_{ji}(n-1) - \beta \Delta \mathbf{v}_{ji}(n-2). \quad (2.57)$$

Krajnji izvodi koji ovdje prikazani za računanje promjena težina za izlazni i sakriveni sloj vrijede jedino za neuronsku mrežu prikazanu slikom 2.3š Error! Reference source not found.te za nju odabrane aktivacijske funkcije sakrivenog i izlaznog sloja. Kao što je vidljivo iz gore izvedenih jednadžbi promjenom aktivacijske funkcije mijenjaju se jedino pripadajuće parcijalne derivacije po funkciji sume neurona, dok su svi ostali parametri identični.

Ukoliko bi mreža sadržavala još jedan ili više sakrivenih slojeva, numerički postupak je u potpunosti analogan postupku za prvi sakriveni sloj, jedino se izrazi dodatno proširuju. Samim unošenjem dodatnih slojeva procesorsko vrijeme računanja se povećava, ali svaki novi sloj donosi određenu kvalitetu u pogledu učenja.

### 2.2.9. Ocjena točnost algoritma učenja

Kako bi se ocijenila točnost, odnosno uspješnost algoritma učenja za određeni zadatak potrebno je definirati mjeru točnosti. Najčešće su u upotrebi tri mjere točnosti učenja: srednja kvadratna pogreška, korijen srednje kvadratne pogreške i normalizirani korijen srednje kvadratne pogreške.

**Srednja kvadratna pogreška MS (eng. Mean Squire error):**

$$MS = \frac{\sum_{n=1}^N (d_n - O_n)^2}{N}, \quad (2.58)$$

gdje je N ukupan broj izlaza, a n broj pojedinog izlaza.

**Korijen srednje kvadratne pogreške (eng. Root Mean Squire error):**

$$RMS = \sqrt{MS} = \sqrt{\frac{\sum_{n=1}^N (d_n - O_n)^2}{N}}. \quad (2.59)$$

**Normalizirani korijen srednje kvadratne pogreške (eng. Normalized Root Mean Squire error):**



$$NRMS = \frac{RMS}{\sigma_{d_n}} = \frac{\sqrt{\frac{\sum_{n=1}^N (d_n - O_n)^2}{N}}}{\sigma_{d_n}}, \quad (2.60)$$

gdje se  $\sigma_{d_n}$ :

$$\sigma_{d_n} = \sqrt{\frac{1}{N} \sum_{n=1}^N (d_n - \bar{d})^2}, \quad (2.61)$$

a  $\bar{d}$  je:

$$\bar{d} = \frac{1}{N} \sum_{n=1}^N d_n. \quad (2.62)$$

U ovom radu je kao mjera točnosti odabran normalizirani korijen srednje kvadratne pogreške odnosno NRMS. Prednost odabrane mjere je njena bezdimenzionalnost koja osigurava neovisnost mjere o dimenzijama učenih veličina i koja omogućuje usporedbu izvedenih algoritama učenja s drugim algoritmima, neovisno o korištenoj sklopovskoj ili programskoj podršci [1].

### 3. Statička unaprijedna neuronska mreža za raspoznavanje osoba

Cilj ovog rada je napraviti statičku unaprijednu neuronsku mrežu koja za zadatak ima raspoznavanje osoba. Samo učenje mreže se odvija na način da se unesu karakteristike osoba te nakon toga da se pusti mrežu da nauči potrebne parametre na način opisan u prethodnom poglavlju te da se potom provedu testiranja rada mreže. Samo unošenje potrebnih karakteristika i pokretanje rada mreže je napravljeno u Matlab GUI sučelju. Proces unošenja parametara i pokretanje samog učenja izvršava se na sljedeći način:

1. odabiranje već unesenih osoba ili odabir novog unosa,
2. unos broja osoba koje se želi unijeti,
3. unos imena pojedinih osoba te unos njihovih karakteristika,
4. unos potrebnih parametara za učenje mreže,
5. učenje mreže,
6. faza testiranja mreže, odabir novih parametara učenja ili izlaz iz samog grafičkog sučelja.

#### 3.1. Topologija mreže

Odabrana mreža se sastoji od ulaznog, jednog sakrivenog i izlaznog sloja. Broj ulaznih neurona u ulaznom sloju je trinaest plus jedan BIAS. Broj izlaznih slojeva ovisi o odabranom broju osoba koje se unose u mrežu te željeni broj osoba unosi korisnik prije unošenja karakteristika osoba. Samim unosom broja osoba se tvori nul-matrica ulaza  $Z$  koja ima 13 stupaca jednakih nuli te četrnaesti stupac jednak jedinici, odnosno 14 ulaznih neurona i onoliko redaka koliko se želi unijeti osoba. Također, unosom broja osoba se tvori matrica izlaza  $D$  koja ima oblik jedinične matrice koja je dimenzija broja osoba te se također tvore izlazni neuroni kojih je također jednak broju osoba koje se želi unijeti u učenje mreže. Broj sakrivenih neurona se zadaje neposredno prije samog pristupanja učenju mreže, a zadaje ga sam korisnik. Nakon unošenja željenog broja sakrivenih neurona, generiranjem slučajnih vrijednosti se tvore matrice  $V$  i  $W$ , gdje je matrica  $V$  matrica težina sakrivenih neurona i ovisna je o broju ulaznih neurona i broju sakrivenih neurona, a matrica  $W$  je matrica težina izlaznih neurona te je ovisna o broju sakrivenih neurona plus BIAS te o broju izlaznih neurona. Također, korisnik mora unijeti i koju aktivacijsku funkciju sakrivenog sloja želi koristiti, želi li koristiti zamah, ako da prvog ili drugog reda te koeficijent zamaha, koeficijent brzine učenja, maksimalni željeni NRMS, na koji način želi provesti učenje te broj koraka nakon kojeg se prekida učenje.

#### 3.2. Karakteristike osoba

Kako bi bilo moguće učenje više osoba uzeto je 10 parametara učenja, odnosno 10 karakteristika osoba. Sve unesene karakteristike se zapisuju i matricu  $Z$ , odnosno u matricu ulaza. Gotovo svaki stupac matrice  $Z$  se odnosi na određenu karakteristiku osobe, međutim, neke karakteristike su opisane sa dva stupca, odnosno dva ulazna neurona. Razlog zbog kojeg su neke karakteristike opisane sa dva neurona će biti objašnjen kasnije tijekom objašnjavanja karakteristika.

Prvo je potrebno unijeti ime osobe za koju se parametri unose. Samo ime se ne sprema u matricu  $Z$  jer Matlab ne podržava pisanje slova, odnosno stringova, različitih veličina u matrice nego, je potrebno imena upisati u ćelije iz kojih se kasnije očitavaju imena osoba. Prvi stupac matrice  $Z$  se odnosi na visinu. Visina se unosi u centimetrima te ju je potrebno normirati između -1 i 1, jer su svi ulazi normirani također između -1 i 1. Normiranje je potrebno kako bi svi neuroni sustava imali jednake šanse utjecati na ishod učenja. Samo normiranje se izvršava nakon unosa svih osoba na način da se izluči prvi stupac matrice  $Z$  te se spremi u novu matricu  $d$ . Potom se iz matrice  $d$  nađu najmanji i najveći iznos visine normiraju između -1 i 1, gdje će najmanja vrijednost visine poprimit iznos -1, a najveći iznos visine će poprimit iznos 1, a ostale vrijednosti će se smjestiti između -1 i 1 u ovisnosti o sljedećim izrazima:

$$r = \frac{\max - \min}{2}, \quad (3.1)$$

$$s = \frac{\max + \min}{2}, \quad (3.2)$$

$$Z(i,1) = \frac{d(i) - s}{r}, \quad (3.3)$$

gdje je  $\max$  maksimalan iznos matrice  $d$ ,  $\min$  je minimalna vrijednost matrice  $d$ , a  $i$  je broj osobe za koju se očitava visina iz matrice  $d$ , odnosno mjesto normiranja te mjesto spremanja u matricu  $d$ . Nakon izračuna normiranih vrijednosti matrica  $d$  se upisuje u prvi stupac matrice  $Z$ .

Sljedeći parametar koji se unosi je masa te se unosi u kilogramima i nalazi se u drugom stupcu matrice  $Z$ . Kao i visinu masu je također potrebno normirati. Sama masa nije vrlo značajan faktor u samom raspoznavanju osoba, iz razloga što je vrlo lako promjenjiva. Odnosno nije joj potrebno pridavati veliku pažnju u smislu svrstavanja između vrijednosti -1 i 1. Zbog te činjenice formula po kojoj se normira masa je:

$$Z(i,2) = \frac{\text{masa}(i)}{200}, \quad (3.4)$$

gdje je  $\text{masa}$  vrijednost mase unesena za osobu  $i$ , a  $Z(i,2)$  mjesto spremanja mase za osobu  $i$ .

Sljedeći parametar se unosi je boja očiju. Sam unos je preko pop-up izbornika te je moguće odabrati jednu od četiri ponuđenih boja ili opciju ostalo ukoliko se tražena boja ne nalazi na popisu. Boje koje je moguće odabrati su: smeđe, sive, zelene, plave i ostale. Boja očiju se također zapisuje u matricu  $Z$ . Međutim, za razliku od visine i mase koje se upisuju u jedan stupac, odnosno zauzimaju jedan ulazni neuron, boja kose se upisuje u dva stupca i zauzima dva ulazna neurona. Razlog uzimanja dva neurona za odabir boje očiju je u tome što prilikom testiranja sa samo jednim neuronom gdje su boje bile normirane vrijednostima: smeđe 1, sive 0.5, ostalo 0, zelene -0.5 te plave -1 nisu pokazale zadovoljavajuće rezultate. Drugim riječima, tako postavljeni pragovi odlučivanja su premali, za toliki broj različitih boja. Da bi se postigla potpuna raznovrsnost, iz tog razloga su uzeta dva ulazna neurona. Tako da su vrijednosti normirane na ovaj način: smeđe [1 1], sive [1 0], ostalo [0 0], zelene [0 -1] te plave [-1 -1] te se u takvom obliku upisuju u matricu  $Z$  u stupce tri i četiri. Ovaj način normiranja je pokazao znatno bolje rezultate te konačno kvalitetnije izvršavanje i testiranje samog programa.

Nakon unošenja boje očiju, unosi se boja kose. Također, boja kose se unosi preko pop-up izbornika te je moguće odabrati jednu od 4 ponuđene boje i jedne opcije ukoliko ta boja nije ponuđena. Boje kose koje su ponuđene su: crna, smeđa, crvena, ostalo te plava. Kako se radi o pet različitih boja, odnosno pet različitih parametara kao i u prethodnom primjeru za unošenje boje očiju tako i za unošenje boje kose je bilo potrebno uzeti dva ulazna neurona zbog kvalitete samog rada i učenja neuronske mreže. Vrijednosti koje su uzete za opis boje kose su: crna [1 1], smeđa [1 0], crvena [0 0], stalo [0 -1] te plava [-1 -1] te se one upisuju u pet i šesti stupac matrice **Z**.

Sljedeći parametar koji je potrebno unijeti preko pop-up izbornika je duljina kose. Duljina kose je definirana kao duga, srednja i kratka. Duljina kose je definirana jednim ulaznim neuronom koji se nalazi u sedmom stupcu matrice **Z**. Vrijednosti koje se unose su: duga 1, srednja 0 te kratka -1.

Nakon unošenja duljine kose, potrebno je unijeti boju obrva. Također se unosi preko pop-up izbornika te je definirana na isti način kao i boja kose s istim bojama i istim načinom normiranja. Razlika je u mjestu upisivanja u matricu **Z**. Boja obrva se upisuje u osmi i deveti stupac.

Sljedeća karakteristika je na redu spol. Spol se također unosi preko pop-up izbornika te se upisuje u deseti stupac matrice **Z**. Ukoliko se radi o ženskoj osobi u matricu **Z** se upisuje 1, a ukoliko se radi o muškoj osobi upisuje se -1.

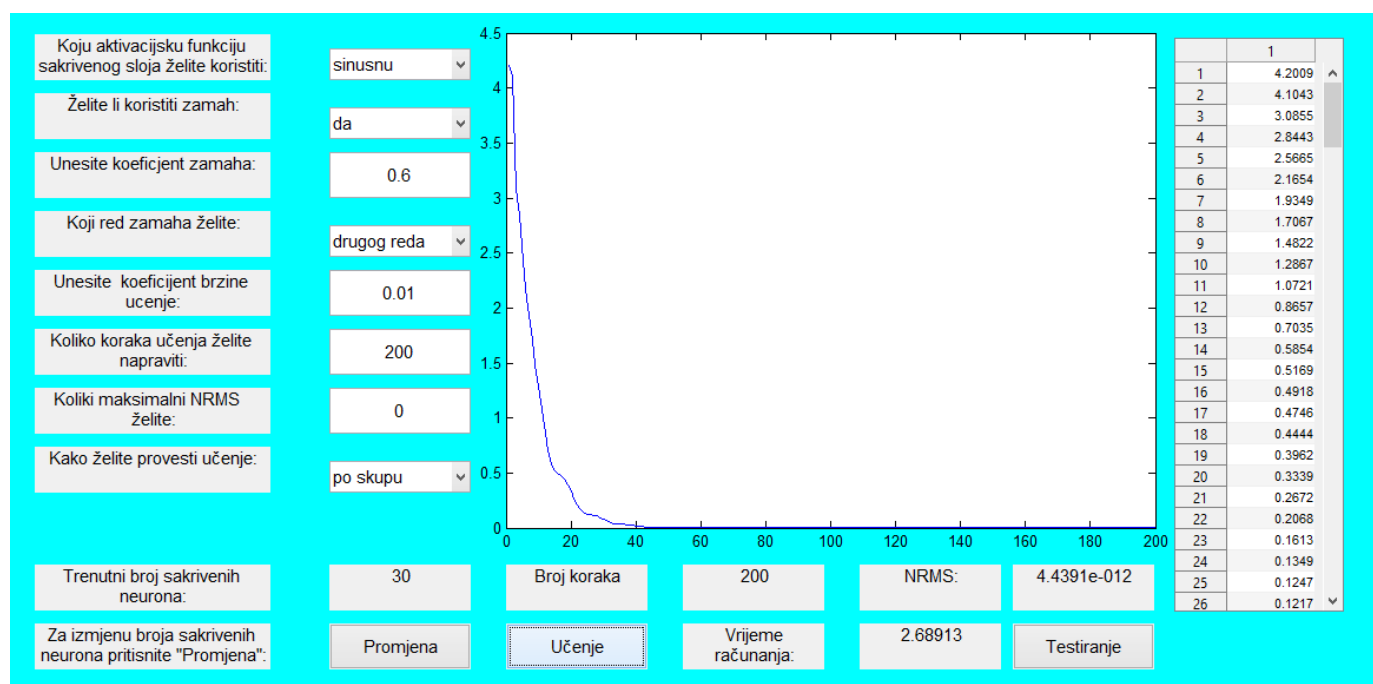
Sljedeće tri karakteristike su: brada, brkovi i nošenje naočala. Svaka od karakteristika ima svoj pop-up izbornik u kojem se odabire da ukoliko osoba ima dotičnu karakteristiku ili ne ukoliko nema. Posjeduje li osoba bradu ili ne, to se upisuje u jedanaesti stupac matrice **Z** te se upisuju vrijednosti 1 ukoliko osoba ima bradu te 0 ukoliko nema. Također jednake vrijednosti se upisuju u matricu **Z** ukoliko se radi o brkovima i naočalama. Razlika je u mjestu upisivanja vrijednosti u matricu **Z**, vrijednosti za brkove se upisuju u dvanaesti stupac, a vrijednosti za naočale se upisuju u trinaesti stupac.

Kako to izgleda u Matlab GUI sučelju prikazuje sljedeća slika:

Slika 3. 1: Prikaz grafičkog sučelja za unos karakteristika osoba

### 3.3. Unošenje parametara učenja i učenje mreže

Sljedeći korak koji je potrebno napraviti je unošenje parametara učenja. Prvo je potrebno unijeti aktivacijsku funkciju sakrivenog sloja. Kao što je bilo objašnjeno u prethodnom poglavlju moguće aktivacijske funkcije su sinusna i nelinearna bipolarna sigmoidalna funkcija. Odabir željene aktivacijske funkcije je preko pop-up izbornika. Kao što je već rečeno na početku ovog poglavlja sljedeće je potrebno unijeti želi li se koristiti zamah ili ne. Ukoliko je odgovor da, pojavljuju se još dodatna dva parametra koje je potrebno unijeti. To je iznos zamaha i koji red zamaha se želi koristiti. Koeficijent zamaha se unosi preko tipkovnice, dok se red zamaha bira preko pop-up izbornika. Potom je potrebno unijeti koeficijent brzine učenja  $\eta$  te se on unosi preko tipkovnice. Nakon unosa koeficijenta brzine učenje potrebno je unijeti maksimalan broj koraka učenja. On je vrlo bitan iz razloga, što će se program sigurno izvršiti u konačnom broju koraka te je to osiguranje ukoliko mreža uđe u beskonačnu petlju i ne uspije pronaći rješenje kojim bi se uspjela zaustaviti u konačnom broju koraka. Potom je potrebno unijeti broj neurona u sakrivenom sloju. Preporuka za unos broja neurona sakrivenog sloja je da ih ne bude manje od neurona u ulaznom sloju, odnosno da ih bude 15 ili više. Nakon unosa broja sakrivenih neurona potrebno je unijeti maksimalni željeni NRMS te na kraju je potrebno odabrati koju vrstu učenja se želi provesti po uzorku ili po skupu. Sljedeća slika prikazuje kako izgleda prozor za unošenja parametara učenja.



Slika 3. 2: Prikaz grafičkog sučelja za unos parametara učenja

Nakon unosa svih parametara učenja pristupa se samom učenju mreže. U samom prozoru koji se otvara prije pokretanja učenja se iscrtava NRMS dijagram, pokazuje se u tablici iznos NRMS-a u svakom koraku, pokazuje broj koraka učenja te procesorsko vrijeme koje je bilo potrebno za izvršavanje učenja.

Također u samom prozoru za učenje je moguće i mijenjati broj sakrivenih neurona pritiskom tipke „Promjena“, te unosom broja sakrivenih neurona u za to predviđeno mjesto. Broj sakrivenih neurona i težine slojeva proizišle iz odabira sakrivenih neurona ostaju konstantni sve dok se ne promijeni broj sakrivenih neurona.

Nakon procesa učenja pristupa se testiranju mreže za unesene osobe u neuronsku mrežu. Testiranje se izvršava preko sličnog prozora kao i za unošenje karakteristika osoba. Jedina razlika je u tome što se ne unosi ime osobe, nego se ime osobe ispisuje na zaslonu nakon unosa svih karakteristika i pokretanja testiranja. Nakon testiranja moguće je odabrati opciju ponovnog upisa parametara učenja ili izaći iz programa.

### 3.4. Testiranje mreže sa 6 osoba

Za testiranje mreže je odabrano šest osoba. U prozoru za testiranje mreže pojavljuju parametri za unos karakteristika, prozor u kojem se ispisuje ime osobe koja odgovara odabranim parametrima te tablica u kojoj su zapisane sve karakteristike svih osoba koje su unesene u mrežu. Kako izgleda prozor za testiranje pokazuje sljedeća slika:

Visina:

Masa:

Boja očiju:

Boja kose:

Duljina kose:

Boja obrva:

Spol:

Brada:

Brkovi:

Naočale:

Pokrenite prepoznavanje

Kraj

	1	2	3	
Ime	Mario	Ana	Kristijan	San
Visina	174	165	180	156
Masa	88	52	70	70
Boja očiju	zelene	smeđe	plave	plav
Boja kose	smeđa	crvena	plava	crve
Duljina kose	kratka	duga	kratka	duga
Boja obrva	smeđe	smeđe	plave	sme
Spol	muško	žensko	muško	žen
Brada	da	ne	ne	ne
Brkovi	ne	ne	ne	ne
Naočale	ne	ne	ne	da

Slika 3. 3: Prikaz grafičkog sučelja za testiranje mreže

### 3.5. Prikaz rezultata različitih načina učenja mreže

Za prikaz rezultata različitih načina učenja mreže bitno je naglasiti da je korištena identična topologija mreže, kao i početne težine u svakom testiranju. Broj sakrivenih neurona u sakrivenom sloju za testiranje je odabran eksperimentom. U testiranju mreže odabrano 30 sakrivenih neurona. U prvoj tablici je prikazano testiranje mreže sa 6 osoba te je korišten koeficijent brzine učenja  $\eta=0,01$  te koeficijent zamaha  $\alpha=0,6$ . Razlog vrlo malog koeficijenta brzine učenja je prikaz rezultata svih kombinacija rada mreže. Odnosno cilj je napraviti usporedbe učenja za sve kombinacije rada mreže u 200 koraka. U drugoj tablici su prikazani rezultati testiranja mreže za koeficijentom brzine učenja  $\eta=0,04$  te koeficijentom učenja  $\alpha=0,6$ .

	Učenje po skupu					
	Sinusna aktivacijska funkcija sakrivenog sloja			Sigmoidalna aktivacijska funkcija sakrivenog sloja		
	Bez zamaha	Zamah prvog reda	Zamah drugog reda	Bez zamaha	Zamah prvog reda	Zamah drugog reda
Vrijeme trajanja	2,61	2,64	2,61	2,05	2,51	2,54
Postignut NRMS	$6,042 \cdot 10^{-7}$	$3,232 \cdot 10^{-14}$	$3,232 \cdot 10^{-15}$	0,03533	0,000162	0,00621
	Učenje po uzorku					
	Sinusna aktivacijska funkcija sakrivenog sloja			Sigmoidalna aktivacijska funkcija sakrivenog sloja		
	Bez zamaha	Zamah prvog reda	Zamah drugog reda	Bez zamaha	Zamah prvog reda	Zamah drugog reda
Vrijeme trajanja	2,65	2,73	2,62	2,83	2,68	2,63
Postignut NRMS	$3,893 \cdot 10^{-7}$	$1,731 \cdot 10^{-12}$	$7,176 \cdot 10^{-11}$	0,0279	$9,584 \cdot 10^{-5}$	$9,584 \cdot 10^{-5}$

Tablica 1: Prikaz rada mreže sa koeficijentom učenja  $\eta=0,01$

Tablica 1 pokazuje rezultate testiranja mreže učenja po skupu i uzorku, sa obje aktivacijske funkcije te sa i bez korištenja zamaha. Iz tablice je vidljivo kako sinusna aktivacijska funkcija sakrivenog sloja postiže manju pogrešku u istom broju napravljenih koraka. Vrijeme izvođenja svakog od načina učenja mreže je približno jednako. Međutim, bitno je naglasiti da se svi rezultati koje prikazuje tablica 1 odnose isključivo na testiranu mrežu te da oni mogu biti drugačiji ukoliko bi se izmijenio samo jedan od parametara mreže ili jedan od parametara učenja. Iz tog razloga nije moguće reći da je jedan od načina najkvalitetniji ili najbrži.



	Učenje po skupu					
	Sinusna aktivacijska funkcija sakrivenog sloja			Sigmoidalna aktivacijska funkcija sakrivenog sloja		
	Bez zamaha	Zamah prvog reda	Zamah drugog reda	Bez zamaha	Prvi red zamaha	Zamah drugog reda
Vrijeme trajanja	2,84	2,54	2,54	2,60	2,71	2,68
Postignut NRMS	$2,39 \cdot 10^{-9}$	$3,168 \cdot 10^{-15}$	$1,387 \cdot 10^{-13}$	$1,02 \cdot 10^{-6}$	$1,713 \cdot 10^{-15}$	$1,713 \cdot 10^{-15}$
	Učenje po uzorku					
	Sinusna aktivacijska funkcija sakrivenog sloja			Sigmoidalna aktivacijska funkcija sakrivenog sloja		
	Bez zamaha	Prvi red zamaha	Zamah drugog reda	Bez zamaha	Prvi red zamaha	Zamah drugog reda
Vrijeme trajanja	2,91	2,85	3	2,82	2,81	2,92
Postignut NRMS	$4,46 \cdot 10^{-16}$	$7,855 \cdot 10^{-12}$	$128 \cdot 10^4$	$3,29 \cdot 10^{-8}$	$7,84 \cdot 10^{-15}$	$2,34 \cdot 10^{22}$

Tablica 2: Prikaz rada mreže sa koeficijentom učenja  $\eta=0,04$

U tablici 2 je pokazano učenje mreže sa povećanim koeficijentom učenja u odnosu na prethodnu tablicu. Iz ove tablice je vidljivo da u slučaju korištenja zamaha drugog reda u učenju po skupu veličina greške raste. Samim time slijedi da učenje ove neuronske mreže korištenjem zamaha drugog reda sa ovim koeficijentom učenja i koeficijentom zamaha nije moguće.

Unaprijedne statičke neuronske mreže sa povratnim prostiranjem pogreške često se upotrebljavaju za rješavanje problema klasifikacije. Tako i u ovom radu se ova mreža koristi za rješavanje problema klasifikacije osoba. Iako ova mreža pokazuje vrlo dobre rezultate, postoji kvalitetnija mreža od ove za ovu vrstu problema. To je RBF mreža, odnosno mreža sa radijalnim baznim funkcijama. je u tome što se njihovo učenje odvija u jednom jedinom koraku, čime se znatno skraćuje vrijeme računanja same mreže.

## 4. Zaključak

Cilj ovog rada je pokazati rad umjetnih neuronskih mreža, njihovo učenje i jednu od mnogobrojnih primjena. U samom radu je korištena troslojna statička unaprijedna neuronska mreža kod koje je u sakrivenom sloju omogućen odabir između dvije aktivacijske funkcije, nelinearne bipolarne sigmoidalne i sinusne. Također omogućen je i odabir broja sakrivenih neurona. U izlaznom sloju neuronske mreže korištena je linearna aktivacijska funkcija, kako bi bilo omogućeno dobivanje rezultata koji su veći od 1 i manji od -1. Način učenja same neuronske mreže povratnim prostiranjem pogreške je s učiteljem. Nadalje, učenje je iterativno te ga je moguće provesti po skupu ili po uzorku. Kako bi se ubrzalo samo učenje, odnosno kako bi se smanjio broj potrebnih iteracija moguća je uporaba zamaha prvog i zamaha drugog reda.

Testiranje izvedenog matematičkog algoritma za učenje statičke unaprijedne neuronske mreže provedeno je na klasifikacijskom primjeru prepoznavanja više osoba, gdje je svaka osoba opisna s nekoliko parametara. Radi se o problemu koji se sastoji od većeg broja ulaznih i izlaznih neurona. Pomoću Matlab matematičkog programskog paketa napravljena je programska podrška, pri čemu nisu korišteni posebni alati za neuronske mreže. Također, pomoću Matlab-ovog alata za izradu grafičkog sučelja je napravljeno grafičko sučelje pomoću kojeg se unose parametri. Samo učenje i unošenje ulaznih parametara je podijeljeno na nekoliko koraka, čime je omogućeno jednostavno korisniku orijentirano korištenje mreže. Pri tome se od korisnika ne zahtijeva poznavanje matematičkog algoritma učenja i testiranja neuronske mreže.

U sklopu samog rada su provedena testiranja mreže za različite parametre učenja. Potrebno je naglasiti da je pri svim testiranjima korišten jednak broj sakrivenih neurona te da su korištene identične početne vrijednosti težina sakrivenog i izlaznog sloja neuronske mreže. Testiranje je vršeno na način da su mijenjane sve ostale vrijednosti potrebne za učenje neuronske mreže. Testiranjima učenja mreže pokazalo se da je u pravilu sinusna aktivacijska funkcija brže postizala željeni iznos maksimalne pogreške, u odnosu na nelinearnu bipolarnu sigmoidalnu aktivacijsku funkciju. Odnosno, korištenjem sinusne aktivacijske funkcije je potrebno manje koraka učenja kako bi mreže naučila zadani problem.

Promatranjem rezultata učenja po skupu i učenja po uzorku, u nešto većem broju slučajeva nešto bolji i brži rezultati dobivani su učenjem po uzorku. Zbog male razlike u rezultatima učenja nije moguće sa sigurnošću tvrditi koji je bolji način učenja neuronske mreže. Iz testiranja se dakle može zaključiti da je za ovu vrstu problema najbolji način učenja za zadane parametre odrediti eksperimentom.

Rezultati učenja korištenjem zamaha prvog reda su pokazali kako mreža bolje i brže uči zadani problem ukoliko se koristi zamah prvog reda. Učenje je moguće ubrzati čak i do nekoliko puta. Međutim, korištenje zamaha drugog reda na ovoj vrsti problema, nije pridonijelo značajnim poboljšanjima učenja, čak u nekim situacijama je usporilo samo učenje u odnosu na učenje u kojem se koristio samo zamah drugog reda.

Kao nastavak ovog rada bi bilo moguće dodati kameru te preko obrade slike prepoznavati karakteristike koje bi mreža učila. Također, moguće je promijeniti ulazne karakteristike pa umjesto prepoznavanja osoba koristiti prepoznavanje različitih predmeta. Integriranjem takve neuronske mreže sa kamerom na robota bi omogućilo robotu na temelju obrađenih informacija da odluči koji sljedeći korak treba napraviti.

## 5. Literatura

1. David Krisel, *A Brief Introduction to Neural Networks*, University of Bonn in Germany 2005.
2. Kenji Suzuki, *Artificial Neural Networks: Architectures and Applications*, InTech, January, 2013.
3. <http://www.science.howstuffworks.com>
4. Branko Novaković, Dubravko Majetić, Mladen Široki, *Umjetne neuronske mreže*, FSB, Zagreb, 2011.
5. P.J.W. Melsa, *Neural networks: A conceptual overview* tellabs Research Center, Mishawaka, IN, USA, 1989.
6. Pierre Peretto, *An introduction to the modeling of neural networks*, Cambridge University press, 1992
7. Guoqiang Peter Zhang, *Neural Networks for Classification: A Survey*, IEEE Transactions on systems, Man, and Cybernetics-Part C: Applications and reviews, Vol. 30, NO. 4, November 2000
8. *Applied History of Psychology*, Wikibooks 2010
9. <http://www.eris.foi.hr>
10. M. Essert, T. Žilić, *Matlab- Matrični laboratorij*, Zagreb, 2004.